GE VERNOVA

PROFICY® SOFTWARE & SERVICES

PROFICY BATCH EXECUTION 5.6

Recipe Development Manual

GE VERNOVA

Proprietary Notice
The information contained in this publication is believed to be accurate and reliable. However, GE Vernova assumes no responsibilities for any errors, omissions or inaccuracies. Information contained in the publication is subject to change without notice.

Trademark Notices

We want to hear from you. If you have any comments, questions, or suggestions about our documentation, send them to the following email address:
doc@ge.com

Table of Contents

About This Guide

The Recipe Development Manual is intended for process control engineers responsible for designing and creating recipes. The manual teaches engineers the skills they need to design recipes with Batch Execution and shows how to implement a recipe strategy.

The manual also assumes that you understand the industrial process, including your process hardware. You should also be familiar with the S88.01 procedural model.

Reference Documents

For related information about developing recipes refer to the following documents:

- ISA-S88.01 Batch Control, Part 1: Models and Terminology
- Equipment Configuration Manual
- Application Guide
- Custom Applications manual
- International Electrotechnical Commission (IEC) 1131-3 specification

Introduction

When using Batch Execution, the Recipe Editor is your management tool for creating and maintaining flexible and reusable recipes.

The Recipe Editor displays recipes graphically and lets you build recipes by pointing and clicking on the tools it supplies, making recipe development quick and easy.

In addition, the Recipe Editor complies with the ISA-S88.01 (SP88) Batch Control standard and supports industry-standard International Electrotechnical Commission (IEC) 1131-3 symbols and sequential function charts (SFC). By using these standards, the recipes you create are easy to understand by other process engineers.

The sections that follow introduce the major concepts and strategies. The goal is to provide a conceptual overview of how the Recipe Editor works and how you can take advantage of its power and flexibility.

If you want to jump right in, go to the Getting Started section. Your Batch Execution software also comes with a set of sample recipes you can examine and use as a guide as you progress through the manual. By default, these recipes reside in the Program Files\Proficy\Proficy Batch Execution\BIN\PROJECTS\DEMO\RECIPES.

Understanding Recipes

Before you begin using the Recipe Editor, review the following sections for a discussion of recipes, sequential function charts, and developing a control strategy:

- What is a Recipe?
- Recipe Types
- Recipe Components

What is a Recipe?

A *recipe* is the information that identifies the following manufacturing requirements of a specific product:

- The necessary ingredients.
- The quantities of each ingredient.
- The production equipment (such as mixers, holding tanks, and ovens) needed to make the product.
- The order in which tasks are performed.

Recipe Types

The S88.01 standard defines the following types of recipes:

- General recipes
- Site recipes
- Master recipes
- Control recipes

For information on recipe types, refer to the Application Guide.

Recipe Components

Master and control recipes consist of the following components:

Header – contains the administrative information for a recipe including the:

- Name
- Version number
- Author
- Version date
- Recipe and product names
- Product code
- Minimum, maximum, and default batch sizes

Parameters – contains a list of process parameters such as mixing time or temperature. Each recipe or step parameter is defined with the Equipment Editor and is associated with a specific piece of equipment (unit) and a phase. Refer to the Equipment Configuration Manual for information on defining process (phase) parameters, and refer to the Understanding Recipe Parameters and Step Parameters section for more information on working with recipe and step parameters.

Equipment requirements – defines a list of the equipment that is required to produce a product. The type of recipe determines the content of the recipe as the following table explains:

When working with...	The recipe contains...
Master recipes	The equipment that can produce the product.
Control recipes	The specific equipment that is used to produce a single batch.

Procedure – defines the process strategy. Master and control recipe procedures are defined using the S88.01 Procedural model, illustrated in the Understanding Sequential Function Charts section. For more detailed information on the Procedural model, refer to the Application Guide.

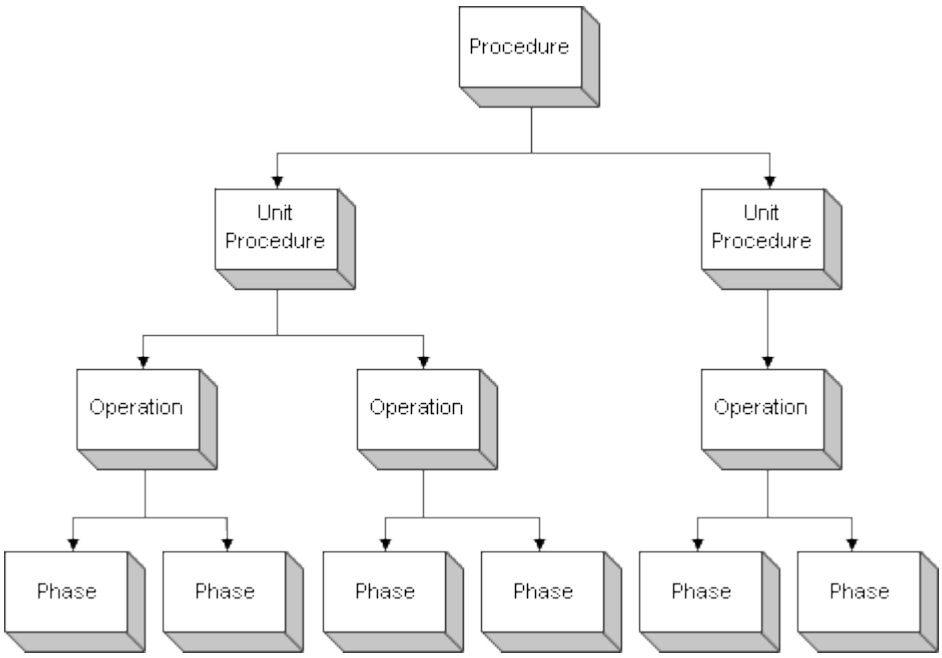
In the Recipe Editor, each procedure, unit procedure, and operation is considered a recipe.

Understanding Sequential Function Charts

The Recipe Editor lets you create procedures using a sequential function chart (SFC) editor. A *sequential function chart* is a graphic representation of a recipe's logic. Each chart uses the S88.01 Procedural model, shown in the following figure, to build a procedure (master recipe).

You can construct a sequential function chart using one or more symbols. These symbols are standardized by the International Electrotechnical Commission (IEC) 1131-3 specification and provide the following benefits to your sequential function charts:

- Unambiguous representation of the recipe's logic.
- Support for decision, loop, and parallel logic structures.
- International recognition.



The S88.01 Procedural Model

For more information about sequential function charts, refer to the Constructing a Sequential Function Chart section. For more information on the S88.01 procedural model, refer to the Application Guide.

Developing a Control Strategy

Before you create recipes for your process, you need to develop a control strategy. This strategy identifies each master recipe you need and the logic of each recipe.

In general, use the following approaches to develop flexible and reusable recipes:

- Build a library of generic operations that you can reuse in multiple recipes.
- Create recipes that can run on multiple units. This type of recipe is called a *class-based recipe*.

For more information on strategies for building recipes, refer to the Implementation Strategies section.

Understanding Active Binding

Batch Execution supports Active Binding™. *Active Binding* encompasses all aspects of binding a unit procedure to a physical unit. The ability to dynamically select a unit on which to run a particular unit procedure allows Batch Execution to bind and re-bind units when a batch is created, as well as during batch production. There are three methods of binding a unit to a unit procedure:

- Manually, when the recipe is created.
- During run time, either using an operator prompt or manually, if the operator has rights to do so.
- Automatically, allowing the Batch Execution Server to make the selection dynamically.

Additionally, the operator may be allowed to change the binding during batch execution if the recipe designer allows it. For more information, refer to the Building Recipes section.

Getting Started

The sections that follow introduce basic Recipe Editor operations and the Batch Execution sample application:

- Before You Start
- Learning Recipe Editor Basics
- Understanding the Sample Application
- Building a Recipe: Overview

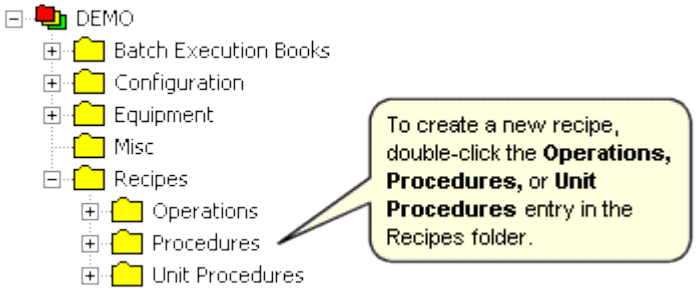
Before You Start

Before you can start creating recipes with the Recipe Editor, make sure the Batch Execution area model is set up with the Equipment Editor. This area model defines the units and phases needed for your recipes.

If you want to take advantage of advanced recipe features, such as phase link groups and equipment capacity, you should consult with the engineer designing the area model. By working closely with the area model designer, you can help to ensure that the area model is designed in a manner that lets you easily use the recipe features you require.

Learning Recipe Editor Basics

Once the area model is ready, you can begin using the Recipe Editor. To start a new recipe with the Recipe Editor, refer to the following figure.



Creating a New Recipe

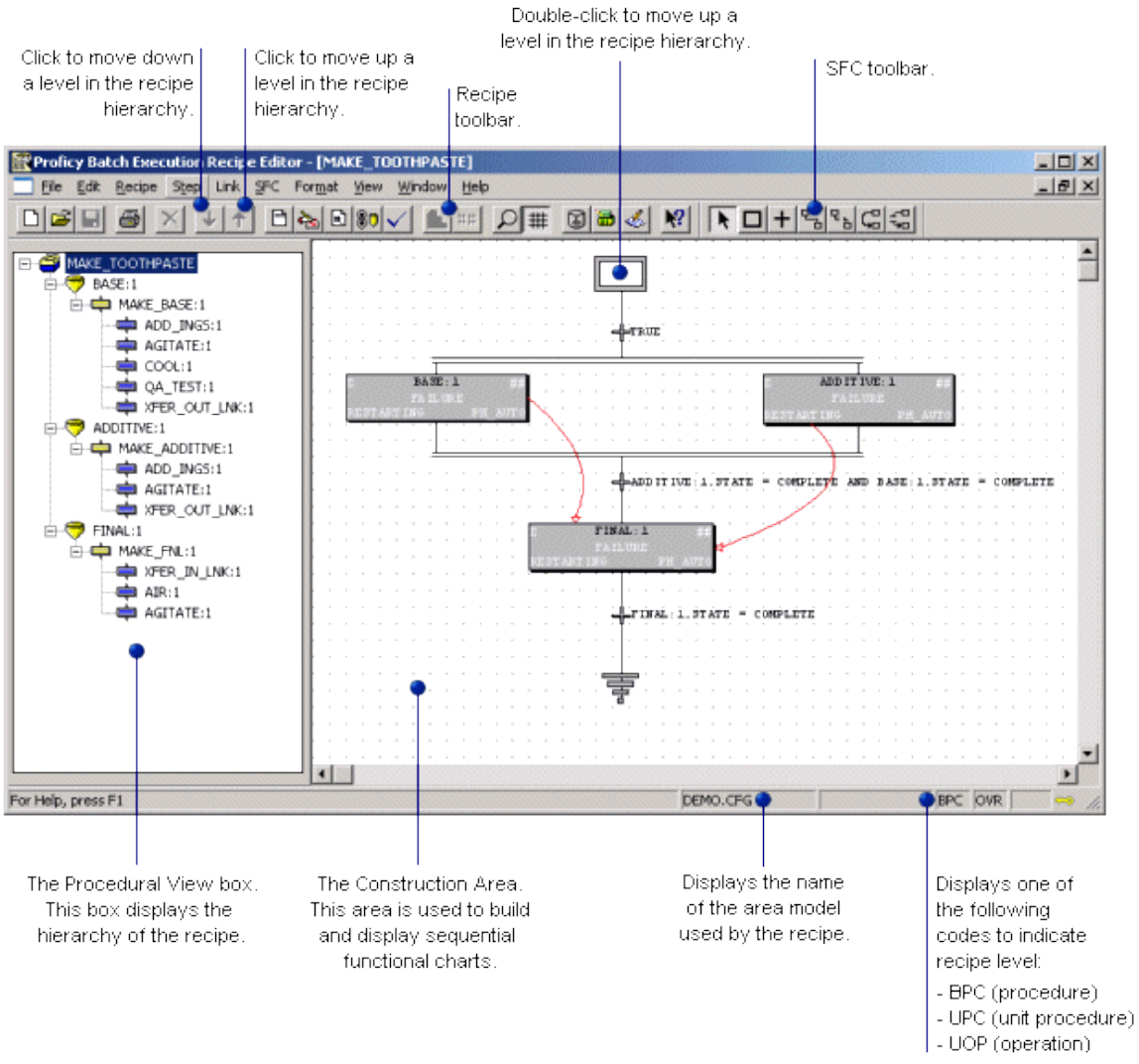
To open an existing recipe, double-click the recipe from the Proficy Batch Execution WorkSpace system tree. The Recipe Editor starts and opens the recipe you selected.

Be aware that if you entered Microsoft SourceSafe information in the Batch Execution Configuration dialog box under the Electronic Work Instruction tab, when you start the Recipe Editor it will try to connect to SourceSafe. This normally takes a few seconds, since the Recipe Editor needs to connect and read down the EIB information. An hourglass displays only for a few seconds if the SourceSafe server computer is running and on the network when you start the Recipe Editor. If SourceSafe is not running, the Recipe Editor displays an hourglass for 2 minutes at startup before it stops trying to connect to SourceSafe and proceed with the startup.

NOTE: *When you are not connected to SourceSafe or if you are not configured to be connected to SourceSafe, the EIB list in the Recipe Editor's Electronic Work Instruction dialog box is empty.*

Displaying a Recipe

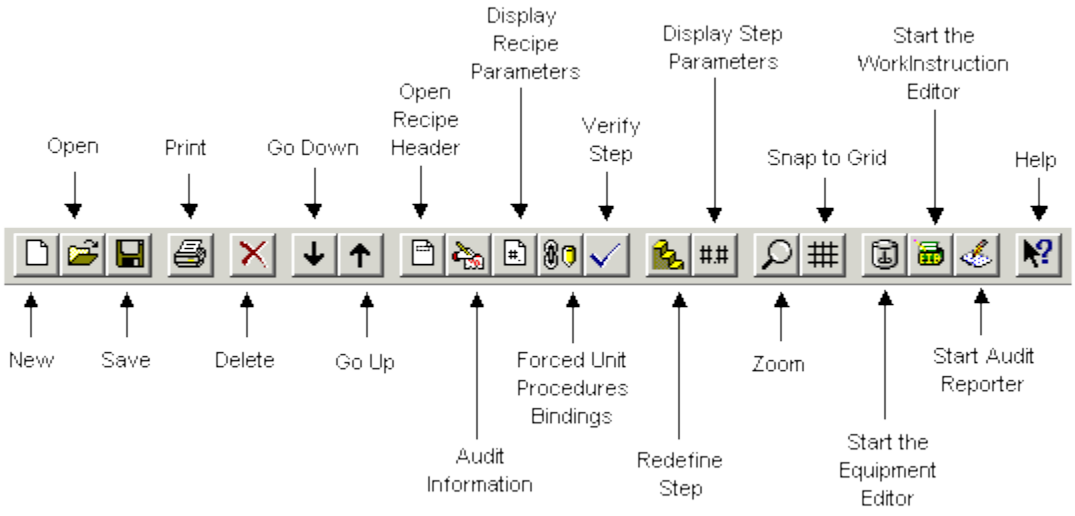
Once the Recipe Editor starts, it opens a window and displays the current recipe, as shown in the following figure.



The Recipe Editor

Using the Recipe Toolbar

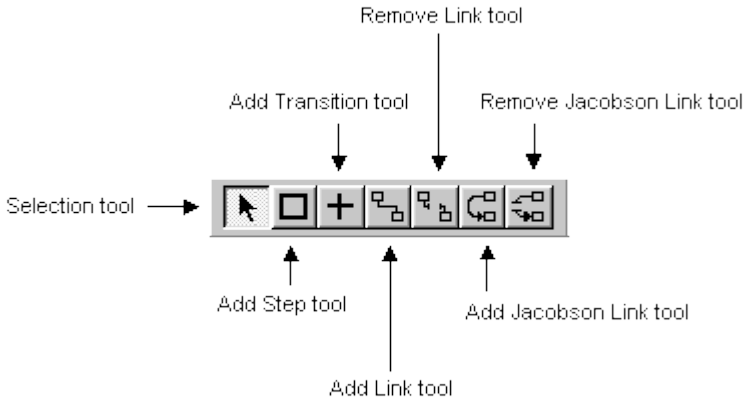
Across the top of the window is the Recipe toolbar. This toolbar, shown in the following figure, provides quick access to common recipe operations such as saving a recipe, verifying a recipe, and displaying recipe parameters.



The Recipe Toolbar

Using the SFC Toolbar

Next to the Recipe toolbar is the SFC toolbar. This toolbar, shown in the following figure, lets you create and modify sequential function charts.



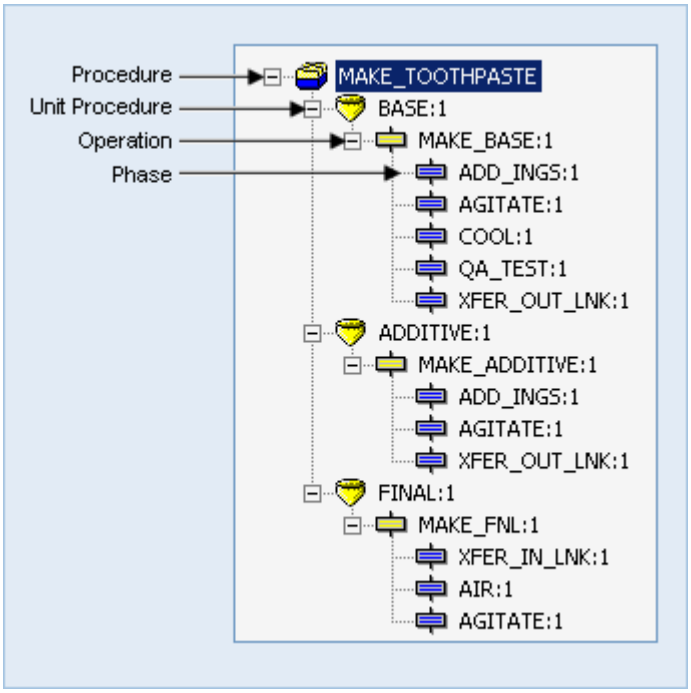
The SFC Toolbar

Navigating the Hierarchical Structure

Each recipe you create appears in the Procedural View box, shown in the following figure. This box displays the hierarchical structure of the current recipe. The exact number of levels displayed depends on the level of the open recipe, as the following table describes.

Displaying the Recipe Hierarchy	
When you open a...	**The Procedural View box displays the...**
Procedure	Unit procedures, operations, and phases defined for the recipe.
Unit Procedure	Operations and phases defined for the unit procedure.
Operation	Phases defined for the operation.

The following image shows examples how the Procedure, Unit Procedure, Operation, and Phase appear in the Procedural View box.



Displaying the Recipe Hierarchy

Expanding and Collapsing the Recipe Hierarchy

Clicking the plus sign to the left side of a step in the Procedural View box opens (expands) the step and displays its contents. Clicking the minus sign closes (collapses) the step and hides its contents.

Navigation Techniques

You can navigate up and down the hierarchy by:

- Selecting a step from the hierarchy. If the selected step is not displayed in the sequential function chart, the Recipe Editor changes to display the appropriate chart.
- Using the arrow buttons on the Recipe toolbar. For example, assume you select the phase XFER_IN_LNK:1 shown in the figure in the Navigating the Hierarchical Structure section. By clicking the Go Up button, the Recipe Editor displays the sequential function chart for the operation MAKE_FNL:1. To re-display the previous chart, select the XFER_IN_LINK:1 step again from the Procedural View box.

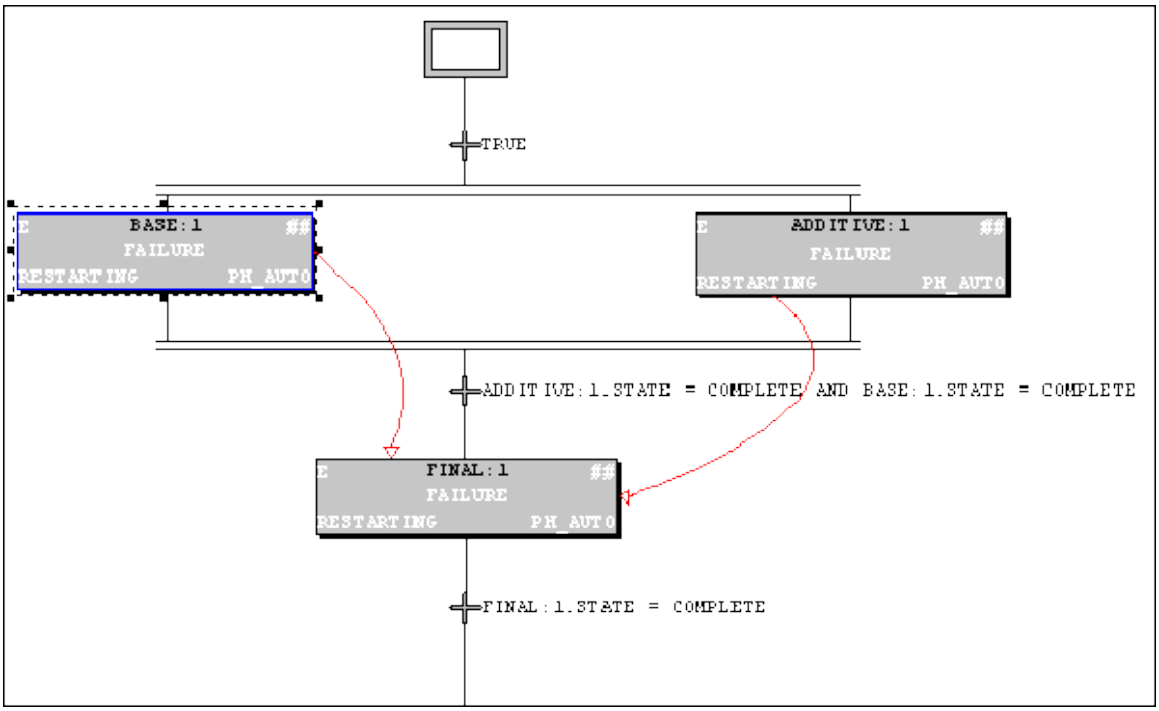
Understanding the Sample Application

This section introduces a sample application for making toothpaste. The purpose of this sample application is to show you how recipes are constructed.

The recipes that constitute the application are provided with your Batch Execution software. You can use these sample recipes as a guide during your recipe development process.

The sample application creates several flavors of baking soda toothpaste. It does this by creating a baking soda base mixture and an additive mixture that are combined together in a reactor. The toothpaste is then packaged in tubes and pump bottles.

The following figure shows the sequential function chart for the sample application to manufacture toothpaste.



Sample Application SFC

Application Features

The sample application takes advantage of the following Batch Execution features that you can incorporate into your recipes:

- Recipe parameters
- Class-based recipes
- Active Binding, including:
    - Forced binding
    - Jacobson Links
    - Unit procedure capacity

Recipe Parameters

Recipe parameters define the amount of each ingredient required for the types of toothpaste in production. By varying the amount of each ingredient, different types of toothpaste are made with the same recipes. For example, to make mint-flavored toothpaste, the recipe parameter MINT is set to 60. To make regular-flavored toothpaste, MINT is set to 0. For information on different strategies using recipe parameters, refer to the Implementation Strategies section.

Class-Based Recipes

Class-based recipes allow the BASE and ADDITIVE unit procedures to use any appropriate mixer defined in the process cell. This feature lets you create recipes that are not tied to a specific piece of equipment. For information on different strategies using class-based recipes, refer to the Implementation Strategies section.

Forced Binding

Forced binding allows you to indicate two unit procedures that either must use the same unit or must use different units. For example, in the sample toothpaste application, the unit procedures BASE and ADDITIVE must use different units. For additional information on forced binding, refer to the Building Recipes section.

Jacobson Links

You can apply Jacobson Links to an SFC in the Recipe Editor to graphically indicate a necessary physical connection between units. For example, the units running the BASE and ADDITIVE unit procedures must both be physically connected to the unit running the FINAL unit procedure. For additional information on the Jacobson Links, refer to the Building Recipes section.

Unit Procedure Capacity

Unit procedure capacity allows you to indicate the amount of material a unit procedure can transfer, process, or contain. By comparing the unit procedure capacity requirements with the defined equipment capacity of units in the area model, the Batch Execution Server can dynamically select an appropriate unit on which to run a particular unit procedure at run time. For additional information on equipment capacity, refer to the Equipment Configuration Manual and the Building Recipes section in this manual.

Building a Recipe

To build a recipe:

1. Click the New button on the Recipe toolbar and specify the type of procedure (operation, unit procedure, or procedure) you want to create. Each procedure level is built independently from all other levels. This modular approach lets you create generic and reusable operations.
2. Define the equipment requirements. This step assigns individual units to the unit procedure or operation you are creating.
3. Build a sequential function chart for the selected procedure level. By creating a chart, you specify the recipe's logic. Use your control strategy to guide you as you build each chart.
4. Define any recipe parameters needed. These parameters override process values defined in the area model.
5. Complete the recipe header. This step is required to verify the recipe. The following fields must be completed:
    - Procedure Identifier (the name of the procedure)
    - Version Number
    - Version Date (the date in this field is automatically generated)
    - Author
6. Verify and save the recipe. Verifying the recipe ensures the recipe is set up correctly. If it is not, the Recipe Editor identifies the problems so you can correct them. =
7. Release the recipe to production.

    NOTE: *A recipe can only be scheduled after it is released to production.*

Constructing a Sequential Function Chart

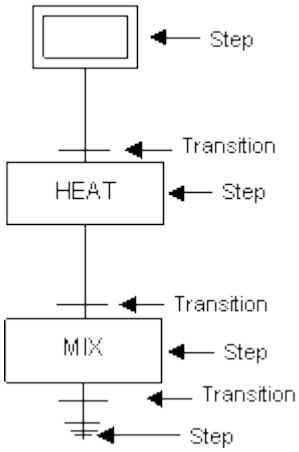
Sequential function charts are the backbone of creating and modifying recipes. The sections that follow discuss the basic tasks needed to create these charts. These tasks include:

1. Adding steps to each operation, unit procedure, and procedure.
2. Configuring each step.
3. Adding transitions.
4. Linking each step and transition together.
5. Configuring each transition.

Once you create a sequential function chart, you need to maintain it so it reflects your changing needs. For information on maintaining a chart, refer to the Maintaining a Sequential Function Chart section.

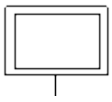
Sequential Function Chart Basics

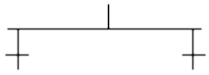
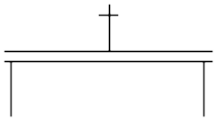
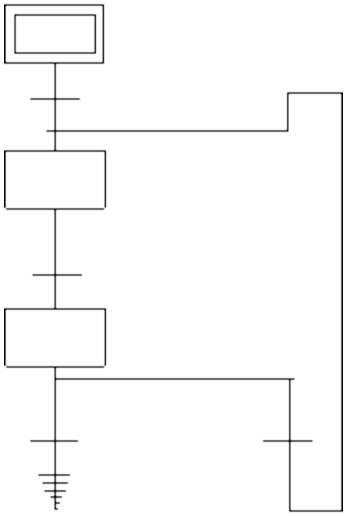
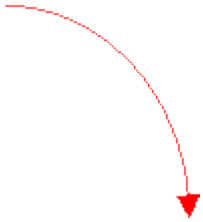
When building recipes, one of your main tasks is to create sequential function charts. *Sequential function charts* are graphical flowcharts that define the logic of a recipe. Each chart consists of steps and transitions that appear in the chart as follows:



Every step (with the exception of the terminal step) must be followed by one or more transitions, and every transition must be followed by one or more steps.

The Recipe Editor uses the symbols shown in the following table to represent different elements in sequential function charts. Each element is discussed in the following subsections.

SFC Graphical Symbols	
This SFC symbol...	**Appears as follows...**
Initial step	
Step	
Terminal step	
Transition	

SFC Graphical Symbols	
This SFC symbol...	**Appears as follows...**
Or structure (decision)	
And structure (parallel execution)	
Loop structure	
Jacobson Link	

NOTE: *The Jacobson Link is shown here because it is a graphical element that can be used in an SFC. For information on Jacobson Links, refer to the Building Recipes section.*

Initial and Terminal Steps

The *initial step* is the logical starting point of a sequential function chart. Similarly, the *terminal step* is the logical end of a sequential function chart. Both steps are required in every chart you make. For this reason, the Recipe Editor automatically adds them to every new recipe you create.

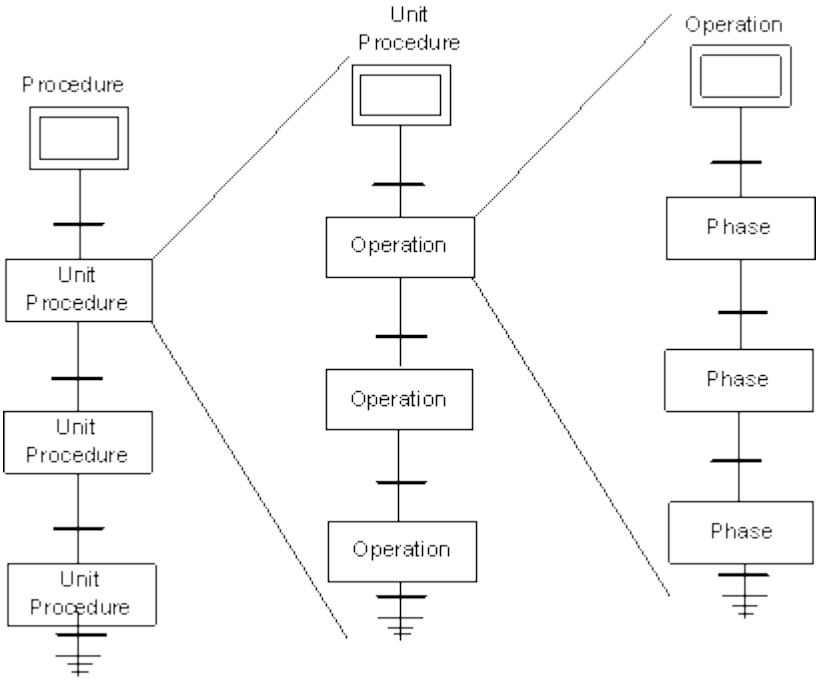
NOTE: *Each recipe has exactly one initial step and one terminal step.*

Recipe Steps

Between the initial step and the terminal step are steps that define the logic of a procedure, unit procedure, or operation. These steps represent an action or another sequential function chart, depending on the recipe level, as the following table shows:

When you create...	The steps in a sequential function chart represent...
A procedure	Unit procedures
A unit procedure	Operations
An operation	Phases

By representing an entire sequential function chart, the Recipe Editor lets you build the procedural hierarchy you need, as the following figure shows.



Sequential Function Chart Interrelationship

When you initially add a step, it is undefined. You can configure an undefined step by selecting the required unit procedure, operation, or phase from a list. This selection provides instructions to Batch Execution on how the recipe should function during production.

Steps can be placed in, and subsequently moved to, any location in a chart.

Transitions

A *transition* defines when a recipe moves from one step to another. Each transition defines a boolean condition that must be true before the step that follows it can run.

By default, the Recipe Editor initially sets each transition you add to a chart to TRUE. However, you do not want a recipe to move onto the next step until the current step is complete or until a specific condition is met. You can edit the default condition by configuring it for your needs.

For additional information on transitions, refer to the Defining Transitions section.

Example: Transition

A typical condition might be as follows:

```
AGITATE.STATE=COMPLETE
```

This condition ensures the AGITATE step is complete before executing the next step in the sequential function chart.

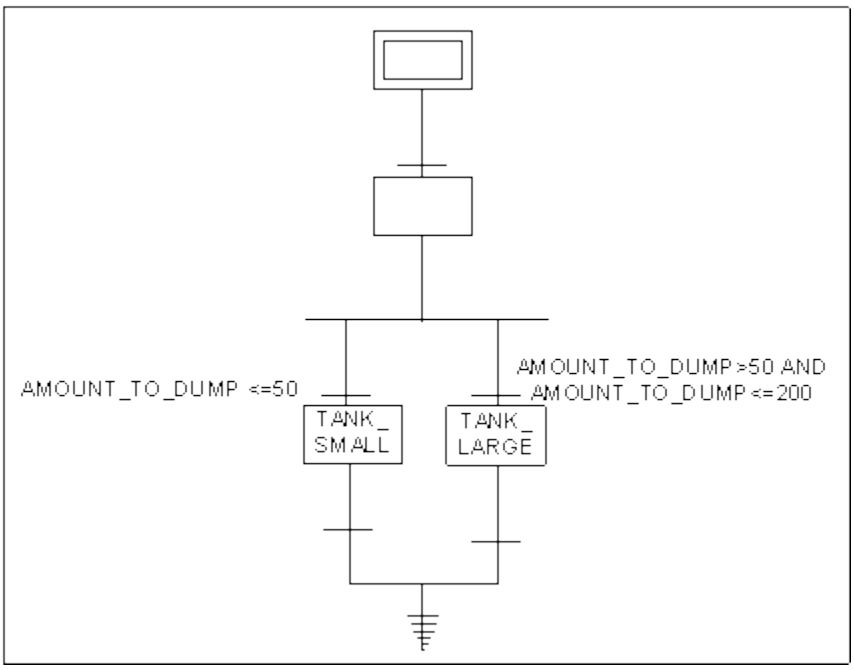
Or Structure

An *Or structure* defines the logic for a decision you want a recipe to make. The result of the decision determines the path that the recipe executes.

Example: Or Structure

For example, suppose an operation can select from two holding tanks. One tank holds up to 50 pounds of material. The other holds up to 200 pounds. Depending on the amount of material processed, you want the operation to select the appropriate size tank.

Using an Or structure lets you set up the logic in the operation so that it can evaluate the amount of material processed. This is done by creating a boolean condition for each transition preceding the step that transfers the material to a specific holding tank. The following figure shows how the sequential function chart for this operation might appear.



Using an Or Structure

Using this chart, the operation can evaluate the amount of material it has to transfer to a holding tank. If the amount is less than or equal to 50 pounds, the TANK_SMALL step runs and transfers the processed material to the smaller tank. If the amount is between 50 and 200 pounds, the TANK_LARGE step runs and transfers the material to the larger tank.

Or Structure Requirements

Only one of the Or structure's conditions are true at any time. As a result, each condition must be mutually exclusive. For example, using the recipe in this figure, you would not want to change the second condition to the following:

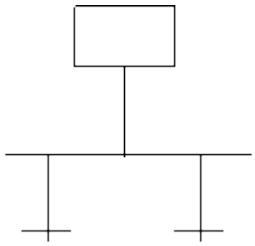
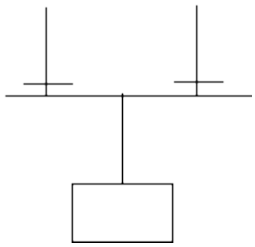
```
AMOUNT_TO_DUMP>=30 AND AMOUNT_TO_DUMP<=200
```

Using this condition, if the amount of material processed is greater than 30 and less than or equal to 50, both conditions would be true. This causes one of the transitions to execute. The exact transition that executes depends on which one evaluates to TRUE first.

Or Convergence and Divergence

You can create two types of Or structures: a divergence and a convergence. An Or divergence connects one step to two or more transitions. An Or convergence connects two or more transitions to one step. Each type of structure is shown in the following table.

NOTE: *An Or divergence must eventually result in an Or convergence.*

Or Structure	
Or Structure	**Example**
Divergence	
Convergence	

And Structure

An *And structure* defines the logic for parallel processing. Use this structure when you need two or more steps to run in parallel.

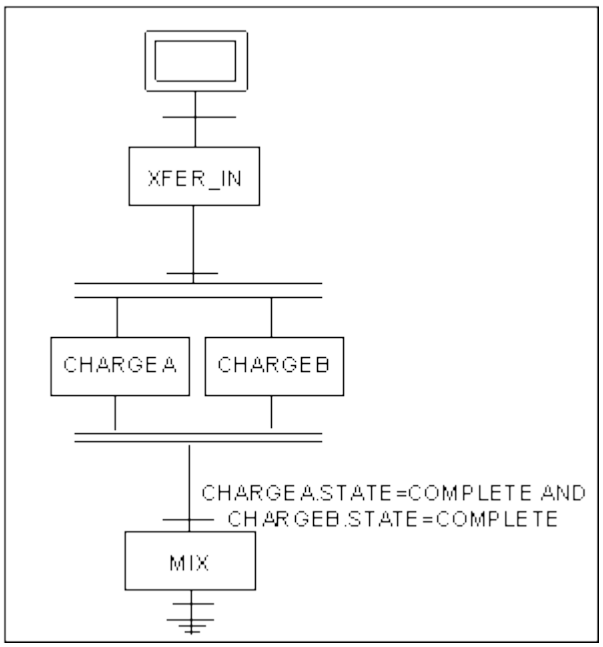
The And structure defines a boolean condition. This condition must be true for the steps that follow it to execute. If part of the condition is true and part is false, the recipe waits until the entire condition is true.

Example: And Structure

For example, suppose a unit procedure needs to charge two vessels at the same time during production. You do not want to move onto the next step (which mixes the contents of the two vessels) until both vessels contain the correct amount of material.

Using an And structure in the operation lets you set up the required logic. This is done by placing a transition after the two CHARGE steps. When the operation runs, the MIX step does not execute until both CHARGE steps are complete.

The following figure shows how the sequential function chart for this operation might look.

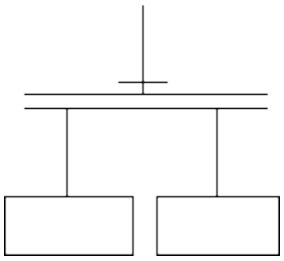


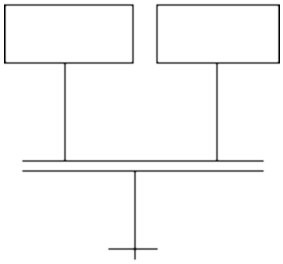
Using an And Structure

And Convergence and Divergence

You can create two types of And structures: a divergence and a convergence. An And divergence connects one transition to two steps. An And convergence connects two steps to one transition. Each type of structure is shown in the following table.

NOTE: *An And divergence must eventually result in an And convergence.*

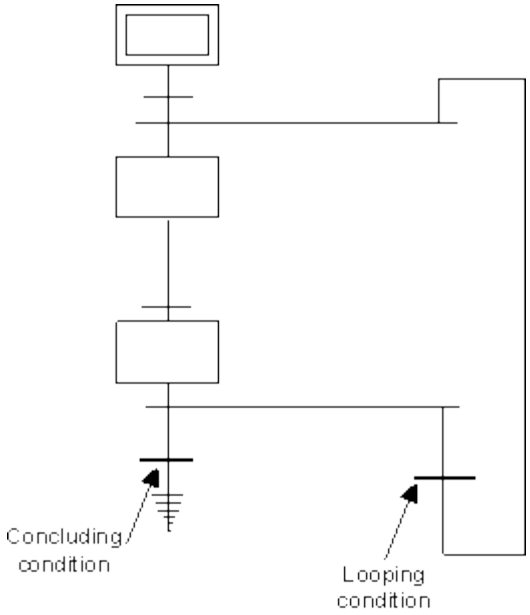
And Structures	
And Structures	**Example**
Divergence	

And Structures	
And Structures	**Example**
Convergence ***NOTE:*** *This transition will not begin to evaluate until all previous steps are active.*	

Sequence Loops

A *sequence loop* defines the logic to repeat a series of steps until a condition is met.

Like the Or structure, you define two boolean conditions, a looping condition, and a concluding condition, as shown in the following figure.



Using a Sequence Loop

Only one of the two transitions in a sequence loop can be true at any time so that the recipe can determine whether or not to loop back or continue on to the next step.

Looping Condition

The looping condition defines the condition for looping back to a previous step. When this condition is true, the recipe goes back to a previous step and executes one or more steps again.

Concluding Condition

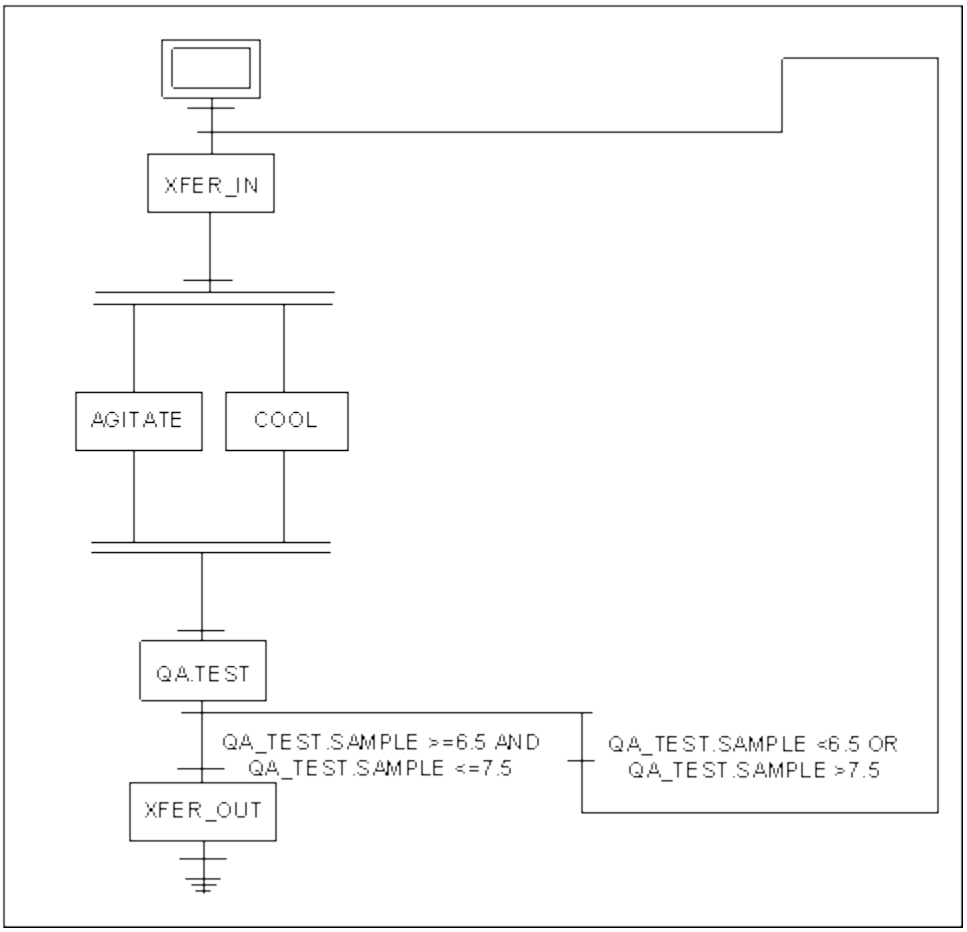
The concluding condition defines the condition that ends the loop and continues to the next step of the recipe. When the looping condition is false and the concluding condition is true, the recipe breaks out of the sequence loop and continues to the next step in the recipe.

Example: Looping Sequence

For example, the sample application needs to do the following tasks while making the toothpaste base:

1. Mix the ingredients for the toothpaste base together.
2. Test the pH level.
3. Add more pH additive if the pH level is too high or too low.
4. Transfer the base to another unit for mixing with the required flavoring.

The following figure shows the sequential function chart representing the operation that handles these tasks.

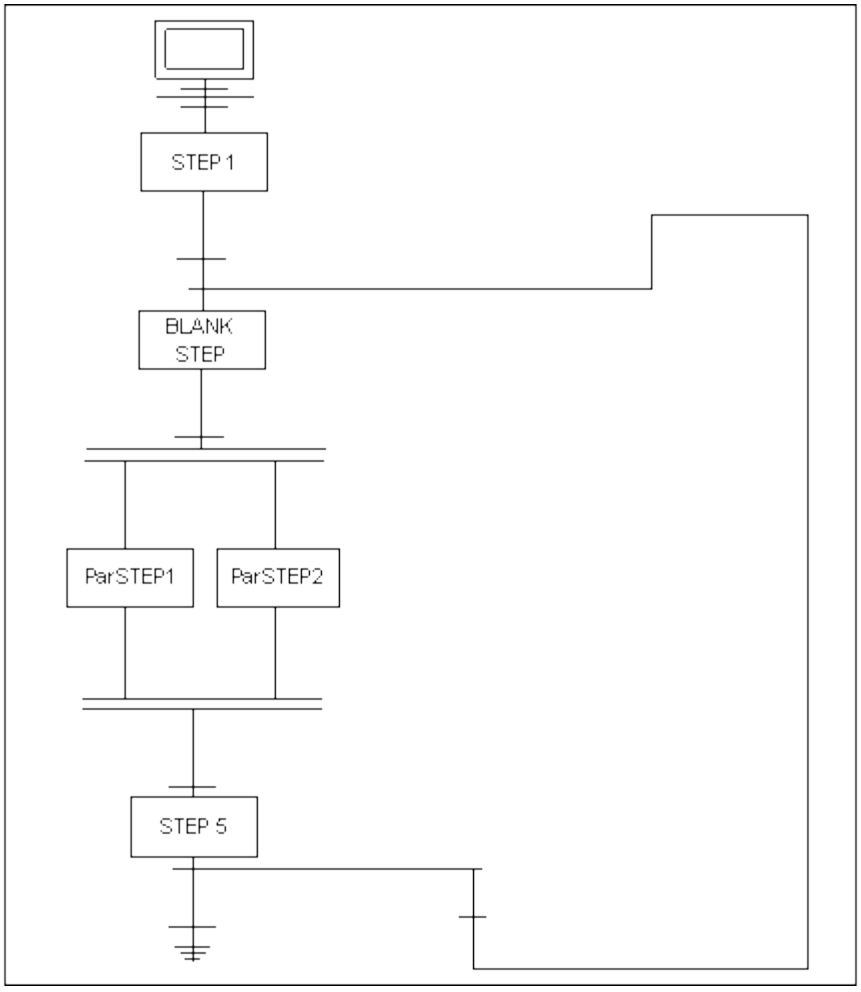


Sample Sequence Loop

When this recipe runs, base ingredients are added to the mixer and then mixed during the AGITATE phase. At the same time, the COOL phase controls the temperature of the toothpaste base.

Once the base ingredients are mixed, a sample is drawn from the mixer to test the base's pH level. The operator is then prompted to enter the pH with the phase parameter SAMPLE. If the value entered is too high or too low, the recipe loops back, adds some pH additive, and continues mixing the base again. Otherwise, the recipe transfers the base to the reactor to be mixed with the required flavoring.

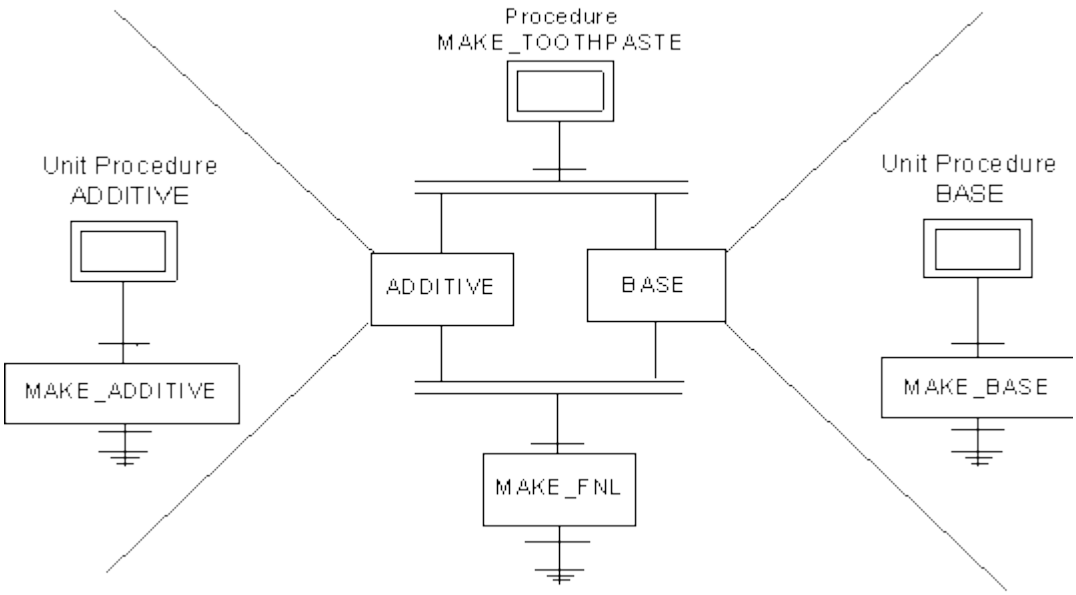
TIP: *It is not possible to loop back to a parallel step; however, to work around this, create a blank (dummy) step and place it above the parallel steps. You can then loop back to the blank step. This is shown in the following figure.*



Using a Blank (Dummy) Step

Understanding How Batch Execution Executes Sequential Function Charts

Before you create sequential function charts, it is important to understand how Batch Execution executes them during production. Because recipes are structured according to the procedural hierarchy, different parts of a recipe can be active at the same time. For example, using the sample application, shown in the following figure, when the step ADDITIVE runs, it starts the unit procedure ADDITIVE. As a result, the MAKE_TOOTHPASTE procedure and the ADDITIVE unit procedure are active at the same time.



Executing Sequential Function Charts

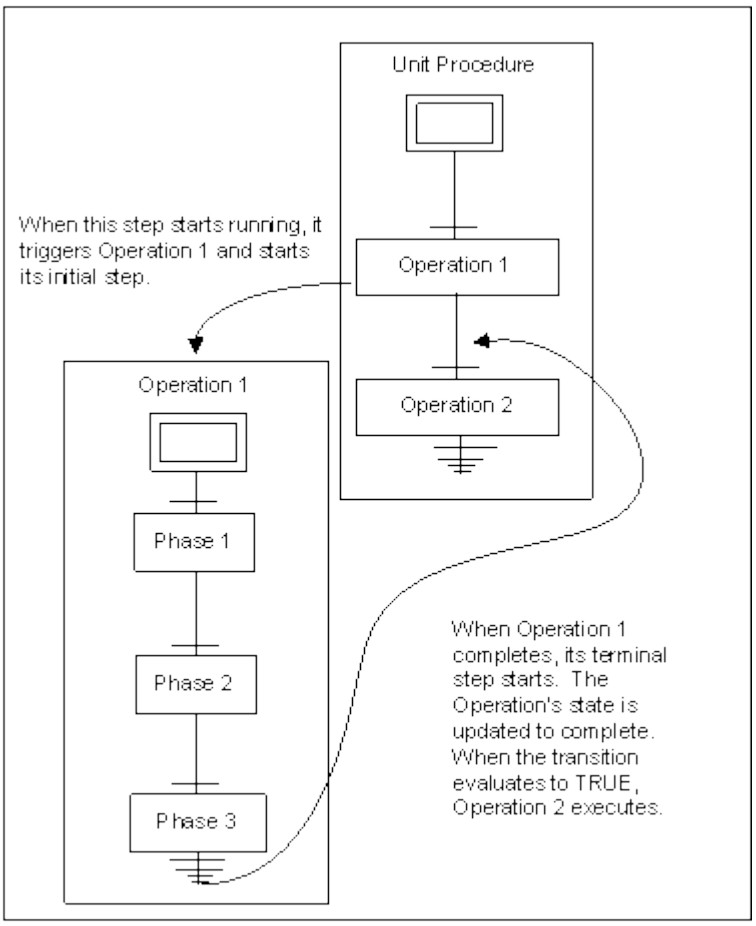
Sequential Function Chart Execution Flow

The basic execution flow of a recipe is as follows:

1. The initial step of the recipe runs and arms its transition.
2. Once armed, Batch Execution monitors the value of the transition.
3. When the transition evaluates to true, Batch Execution stops the previous step and starts the next step.
4. Once the next step starts, the previous transition is reset and the next transition is armed.
5. Steps 2 through 4 repeat until the entire recipe has been executed.

This basic flow is repeated at every level of the procedural hierarchy with the following additions:

- If the current step is another recipe, then when it starts running, execution passes down one level of the procedural hierarchy, and the recipe at that level starts running (starting with the initial step).
- For example, using the sample application, when the step ADDITIVE runs, execution passes down to the unit procedure ADDITIVE and the first step of the unit procedure starts to run.
- When there are no more steps in a recipe at a specific level, the terminal step starts, as the following figure shows. The terminal step behaves much like any other step; however, when the terminal step starts, the higher level step's state becomes Complete.



Passing Control Between Sequential Function Charts

- If two steps follow the currently armed transition, both steps execute simultaneously when the transition evaluates to true. Thereafter, each branch executes as previously described.

  NOTE: *The higher level procedure always has control. If the transition becomes TRUE before the operation completes, then the unit procedure forces the operation to stop and the unit procedure continues execution.*

Example: Understanding Sequential Function Chart Execution

Consider what happens when the sample application runs:

1. A command is sent from the Batch Execution Client to the Batch Execution Server to start the MAKE_TOOTHPASTE recipe.
2. The initial step of the procedure runs and arms its transition.
3. Once armed, Batch Execution monitors the value of the transition. The transition evaluates resource needs for the downstream procedural steps and acquires any necessary resources. Because this transition is TRUE, the initial step stops running and the BASE and ADDITIVE steps in the AND structure start running simultaneously.

4. Batch Execution then examines each step to determine if either step is a recipe.
5. Because both steps are recipes, Batch Execution passes execution down to the unit procedures BASE and ADDITIVE. As a result, the initial steps of these recipes start running and their transitions arm.
6. Once armed, Batch Execution monitors the values of these transitions. Because both transitions are TRUE, the initial steps stop running and the MAKE_BASE and MAKE_ADDITIVE steps start running.
7. Batch Execution then examines each step to determine if either step is a recipe.
8. Because both steps are recipes, Batch Execution passes execution down to the operations MAKE_BASE and MAKE_ADDITIVE. As a result, the initial steps of these recipes start running and their transitions arm.
9. Each phase in these operations executes.
10. When both unit procedures are complete, the transition following these steps evaluates to TRUE and the MAKE_FNL step starts running.
11. The recipe continues as described until the last phase completes. The recipe is complete when its terminal step runs.

Executing Or Structures and Sequence Loops

It is possible for a step to have more than one transition following it. This occurs when Or structures or sequence loops are defined in a sequential function chart. Or structures and sequence loops execute as follows:

1. A step with multiple transitions starts running.
2. Batch Execution arms each transition following the current step.
3. Once armed, Batch Execution monitors the value of each transition.
4. When one of the transitions evaluates to true, Batch Execution stops the step that is running and starts the next step.
5. All the previous transitions are reset and the next transition is armed.

Example: Executing Or Structures and Sequence Loops

For example, the sequence loop in the example application, shown in the following figure, tests the pH level of the toothpaste base and executes as follows:

1. When the phase QA_TEST runs, Batch Execution arms each transition following it.
2. Once armed, Batch Execution monitors the value of each transition.
3. The recipe prompts an operator to enter a value for the phase parameter, SAMPLE.
4. Once a value is entered, Batch Execution evaluates each transition to determine which one is true.

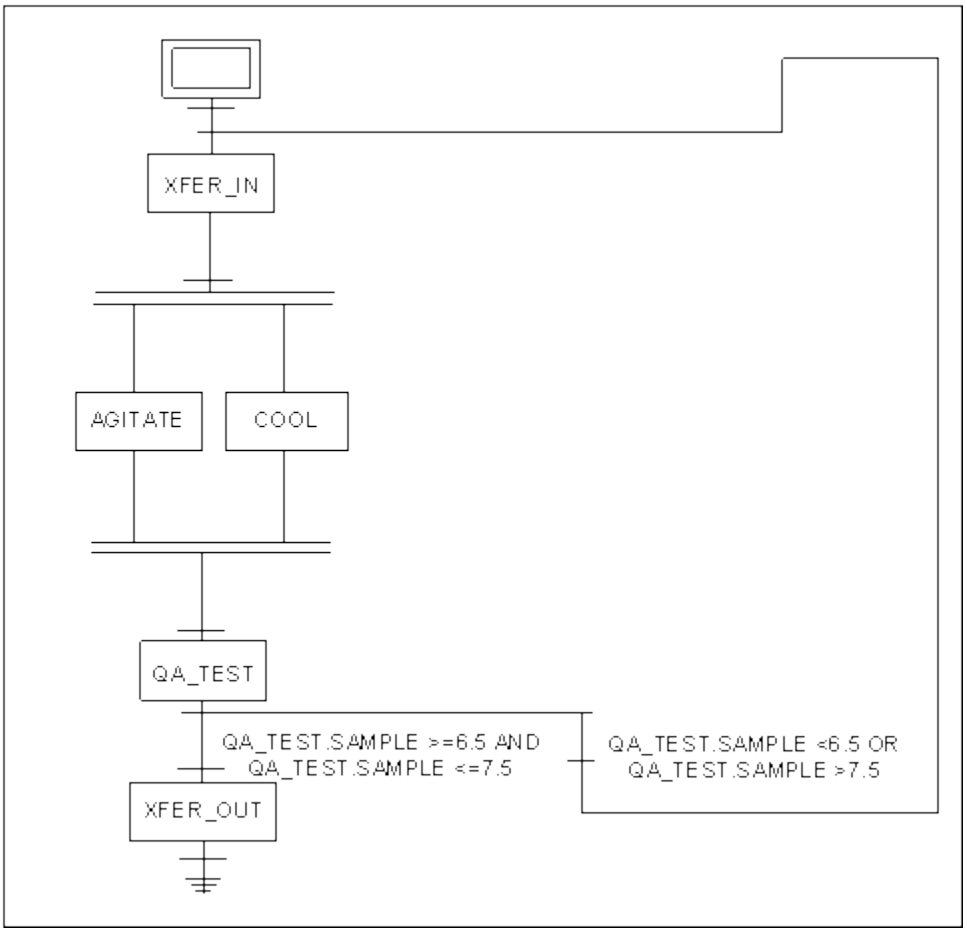
   If the value entered is less than 6.5 or greater than 7.5 then:

   a. The QA_TEST phase stops.
   b. The XFER_IN phase runs again so that more pH additive can be added to the toothpaste base.

      If the value entered is between 6.5 and 7.5 then:

c. The phase is complete.

d. The XFER_OUT phase runs.



Executing a Sequence Loop in a Sequential Function Chart

Creating Steps

You can create a step for a sequential function chart quickly and easily using the Step tool. This tool lets you add steps to a sequential function chart by doing the following:

1. Selecting the Step tool.
2. Positioning the mouse where you want the step to appear.
3. Clicking the mouse button.

Each step that you add is undefined and takes no action until you configure the step. By configuring it, you assign a phase, an operation, or a unit procedure to the step. The exact assignment you make depends on the level of recipe you are creating.

NOTE: *You cannot create or configure initial steps and terminal steps. The Recipe Editor automatically adds and configures them for you.*

Defining Transitions

You must place a transition between each step. The transition defines when a recipe stops executing one step and starts executing the next step. Transitions, therefore, control the execution of the recipe. The movement from one step to another is controlled by the boolean condition of the transition. When the condition evaluates to true:

1. The preceding step stops running.
2. The step that follows the transition begins running.
3. Resource allocation for the following step occurs.

Example: Transition Syntax

For example, the following conditions can be defined in a transition:

- MIX1.MIX_TIME = "50"
- ADDITIVE:1.STATE = COMPLETE AND BASE:1.STATE = COMPLETE

Syntax: Defining Transitions

Transitions use the following syntax:

```
Value

Value operator value
```

where *value* is a constant or an identifier.

The boolean condition in transitions can contain the following elements:

- Constants
- Identifiers
- Operators

More information about these elements is available in the following sections: Understanding Constants, Understanding Identifiers, and Understanding Operators.

Understanding Constants

You can include the following constants as a boolean condition in a transition:

- FALSE
- TRUE
- Any decimal or integer number
- A string enclosed in quotation marks

When using a real or integer number in a boolean condition without an operation, non-zero values are evaluated as true and 0 is evaluated as false.

Example: String Constants

Strings used in a boolean condition require an operator because they cannot be evaluated to true or false. Instead, strings can only be compared to another value as the following examples show:

```
"MINT" = MIX1.FLAVOR

MIX1.MIX_TIME = "50"
```

Understanding Identifiers

You can include one or more identifiers in a boolean condition. Each identifier represents a value. When Batch Execution evaluates the identifier's value in a condition without an operator, non-zero values are evaluated as true and 0 is evaluated as false. The available identifiers are:

- Step attributes
- Step parameters
- Report parameters
- Recipe parameters
- Tags and units

Understanding STEP Attributes

Associated with each step is a set of attributes. Each attribute provides information about the step, such as the state of the step or whether the step has encountered an error. By testing the value of an attribute, a recipe can proceed to the next step when the condition evaluates to true. This is commonly done to determine if the current step is complete.

Syntax: STEP Attributes

The syntax for specifying a step attribute in a boolean condition is as follows:

```
stepname.attribute
```

Example: STEP Attributes

For example, to determine if the step MIX1 is complete, use the following boolean condition:

```
MIX1.STATE = COMPLETE
```

The following table lists the step attributes available in the Recipe Editor.

Recipe Editor Step Attributes	
Attribute	**Syntax**

STATE – defines the state of the step. All states are evaluated as non-zero values. As a result, when this attribute is used without an operator, Batch Execution automatically evaluates it as true. To avoid this evaluation, always use an operator with the STATE attribute. Possible states are: Aborting, Holding, Stopping, Restarting, Running, Held, Complete, Stopped, Aborted, and Idle.
For example, to test if the step MIX1 is running, use the condition:
MIX1.STATE = RUNNING
	`stepname.STATE`
	or
	`stepname.STATE operator state`

FAILURE – indicates whether the step encountered an error. When no error is encountered, the attribute's value is 0. However, when the attribute's value is non-zero, an error occurred.
For example, to test if the step MIX1 encounters an error, use the condition:
MIX1.FAILURE = 1
	`stepname.FAILURE`
	or
	`stepname.FAILURE operator value`

OWNERID – specifies whether a recipe or an external source controls the current step. When the recipe controls the step, the attribute's value is 0. When an external source controls the step, the attribute's value is 1.
For example, to test if the step MIX1 is controlled by the recipe, use the condition:
MIX1.OWNERID = 0
	`stepname.OWNERID`
	or
	`stepname.OWNERID operator value`

PAUSE – indicates whether the current step is pausing. Possible values of the attribute are TRUE or FALSE.
For example, to test if the step MIX1 is pausing, use the condition:
MIX1.PAUSE = TRUE
	`stepname.PAUSE`
	or
	`stepname.PAUSE operator value`

Recipe Editor Step Attributes	
Attribute	**Syntax**

PAUSED – indicates whether the current step is paused. Possible values of the attribute are TRUE or FALSE.
For example, to test if the step MIX1 has paused, use the condition:
MIX1.PAUSED = TRUE
	`stepname.PAUSED`
	or
	`stepname.PAUSED operator value`

SINGLESTEP – indicates the phase logic in the PLC is in single step mode. This mode lets the step pause at each pause location. A pause location is a preprogrammed pause in the phase logic. Typically, the phase logic is in single step mode when testing a phase. Possible values of the attribute are TRUE or FALSE.
For example, to test if the step MIX1 is in single step mode, use the condition:
MIX1.SINGLESTEP = TRUE
	`stepname.SINGLESTEP`
	or
	`stepname.SINGLESTEP operator value`

STEPINDEX – defines the step index number of the pause location being executed in the phase logic.
For example, to test if the step MIX1 has reached step index pause location 5, use the condition:
MIX1.STEPINDEX >= 5
	`stepname.STEPINDEX`
	or
	`stepname.STEPINDEX operator value`
	Possible values are integers greater than zero. As a result, when this attribute is used without an operator it is automatically evaluated as true. To avoid this evaluation, always use an operator with the STEPINDEX attribute.

Understanding Step and Report Parameters

You can include step parameters, report parameters, and recipe parameters in the boolean condition of a transition. Step and report parameters are variables that represent process values and are defined in the area model. Each step or report parameter is associated with a phase. Step parameters in a recipe correspond to phase parameters in the area model. Phase parameters represent process values used by the phase while the phase is running, such as the mixing time for a batch or the amount of material to add to a mixture. Report parameters, by comparison, are uploaded from the process hardware. Typical report parameters are the actual mixing time or the actual amount of material added. Refer to the Equipment Configuration Manual for information on configuring phase and report parameters.

Understanding Recipe Parameters

Recipe parameters are like step parameters. They are variables that represent process values. The difference between step parameters and recipe parameters is you create recipe parameters in a procedure, unit procedure, or operation. Once created, you can use a recipe parameter to override the default value of a step parameter. For more information about recipe parameters, refer to the Understanding Recipe Parameters and Step Parameters section

Understanding Step Parameters

Individual steps in a sequential function chart can also have parameters. Step parameters are always defined at the next level down in the procedural hierarchy.

Example: Step Parameters, Report Parameters, and Recipe Parameters

Typically you include a step parameter, report parameter, or recipe parameter in a transition when proceeding to the next step depends on the value of a parameter. For example, suppose you need to mix one group of ingredients for 10 minutes before adding another group of ingredients. Assume the phase MIX1 controls the mixing and the phase parameter MIX_TIME defines the time interval. To determine when to add the next group of ingredients, you might use the following transition:

```
MIX1.MIX_TIME = 10
```

Now assume you create a recipe parameter, MIXING_TIME, for the operation that uses the phase MIX1. You can create a transition with the recipe parameter using the following boolean condition:

```
MIXING_TIME = 10
```

If the step parameter and the recipe parameter are associated with two different mixers running in parallel, you could create a boolean condition similar to the following:

```
MIX1.MIX_TIME = MIXING_TIME
```

This condition ensures that the mixing time of the two mixers is the same before continuing with the recipe.

Syntax: Step Parameters, Report Parameters, and Recipe Parameters

When you include a step parameter, report parameter, or recipe parameter in a boolean condition, use one of the following:

```
stepname.parametername
stepname.parametername operator value
```

When you include a recipe parameter in a boolean condition, use one of the following:

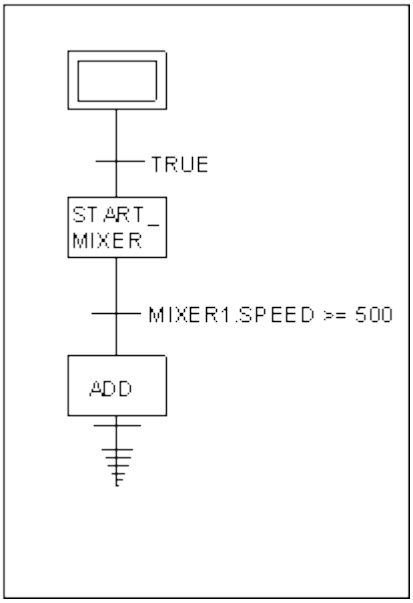
```
recipeparametername
recipeparametername operator value
```

Working with Tags and Units

You can include a unit and tag in a boolean condition. A *unit* is a device defined in the area model. Each device has physical characteristics, such as the speed of the motor, the size of the tank, and so on. To represent these characteristics, tags are defined for each unit.

Example: Tags and Units

You can specify a unit and tag in a boolean condition whenever moving from one step to another depends on a unit characteristic. For example, suppose an operation requires the speed of the mixer to be running at 500 RPM or higher before adding any ingredients. Also assume the unit is MIXER1 and the tag associated with the speed of the mixer is SPEED. The following figure shows the logic for this operation.



Using Tags and Units

With this operation, the mixer starts during the START_MIXER step. When the speed of the motor is greater than 500 RPM, ingredients are added by the ADD step.

Syntax: Tags and Units

When you include a unit and tag in a boolean condition, use one of the following:

```
unitname.tagname
unitname.tagname operator value
```

Understanding Operators

The operators in a boolean condition are symbols that let you connect constants and identifiers together to create boolean expressions. You can include the following types of operators:

- Mathematical operators
- Relational operators
- Boolean operators

In addition to these operators, you can include parentheses to group parts of a boolean condition together. For more information on grouping parts of a boolean condition, refer to the Changing the Order of Precedence section.

Mathematical Operators

Mathematical operators let you add, subtract, multiply, and divide two values. Using these operators, you can create a mathematical expression for a transition.

Mathematical expressions are evaluated by calculating the expression's value. If the resulting number is 0, the expression evaluates to FALSE. If the resulting number is non-zero, the expression evaluates to TRUE.

Generally, mathematical expressions by themselves are not used as a boolean condition. Instead, mathematical expressions are included as part of a larger condition using relational operators.

Syntax: Mathematical Operators

The syntax for each mathematical operator is provided in the following table.

Mathematical Operator Syntax	
Operator	**Syntax**
+ (addition)	*value+value*
- (subtraction)	*value-value*
* (multiplication)	*value*value*
/ (division)	*value*/*value*

Example: Mathematical Operators

Consider the following expressions:

Expression	**Value**
3+1	4
1+-1	0

Relational Operators

Relational operators let you compare two values to determine if the values are the same or to determine if one value is greater than another.

Syntax: Relational Operators

The syntax for relational operators is shown in the following table.

Relational Operator Syntax	
Operator	**Syntax**
= (equal to)	*value=value*
<> (not equal to)	*value<>value*
> (greater than)	*value>value*
< (less than)	*value<value*
>= (greater than or equal to)	*value>=value*
<= (less than or equal to)	*value<=value*

Example: Relational Operators

Relational operators are commonly used in boolean conditions to determine if a recipe parameter, step parameter, or step attribute has a specified value. A common example of this is:

```
MIX1.STATE = COMPLETE
```

When Batch Execution evaluates this condition, the value on the left side of the operator is compared to the right side. If the two values match, the condition is true. Otherwise, it is false.

You can also compare any constant or identifier to any other constant or identifier. For example, consider the following boolean conditions:

Condition	**Value**
3>1	TRUE
MIXER1.SPEED*5>=500	TRUE, if MIXER1.SPEED is 100 or more.
MIX1.MIX_TIME<=10	TRUE, if MIX1.MIX_TIME is less than or equal to 10.

Boolean Operators

Boolean operators let you connect two or more boolean conditions.

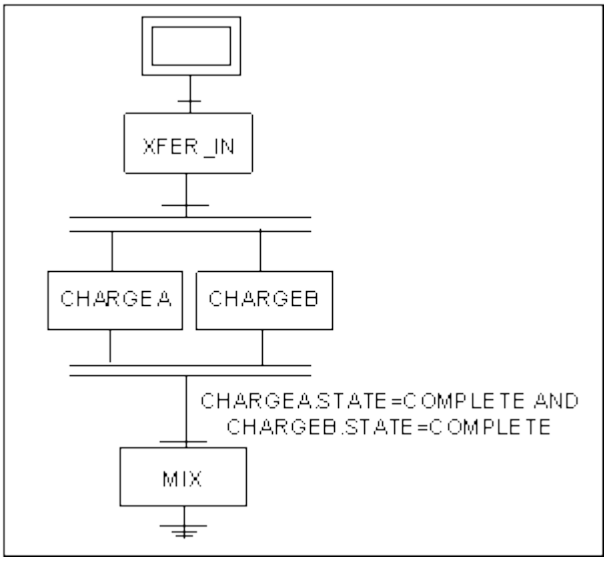
Syntax: Boolean Operators

The syntax for each operator is listed in the following table.

Boolean Operator Syntax	
Operator	**Syntax**
AND	*condition* AND *condition*
OR	*condition* OR *condition*
NOT	NOT *condition*

Example: AND Operator

Depending on the structure of your sequential function chart you may need to use AND operators. For example, assume you need to charge two sets of ingredients simultaneously and then mix them. To accomplish this, you create a recipe with an AND structure as follows:



The transition following the AND structure uses the AND operator. When this transition is evaluated, the expressions on both sides of the operator must be true for the entire condition to be true. If either side is false, the entire condition is false. By using the AND operator, you ensure both steps are complete before proceeding to the next step.

Example: OR Operator

Use the OR operator to evaluate if one of two or more conditions are true. For example, suppose you need to mix ingredients for 5 minutes or until the mixture is heated to 100 degrees. If the step parameter ACTUAL_MIX_TIME reports the actual time that the ingredients are mixed and the phase parameter ACTUAL_TEMP reports the actual temperature at which the mixture was heated, then the boolean condition for this transition would be:

```
MIX:1.ACTUAL_MIX_TIME >= 5 OR ACTUAL_TEMP >= 100
```

Example: NOT Operator

Unlike AND and OR, the NOT operator does not connect two boolean conditions; it inverts the value of a condition. For example, assume the following condition is true:

```
MIX1.STATE = COMPLETE
```

Then the following condition is false.

```
NOT MIX1.STATE = COMPLETE
```

NOT Operator in an Or Structure

One common way of using the NOT operator is with an Or structure. One requirement of the Or structure is that only one of its boolean conditions can be true at any time. As a result, each condition must be mutually exclusive. You can ensure that each condition is mutually exclusive by using the NOT operator.

When using the NOT operator, use the syntax for one transition:

```
STEPA.FAILURE
```

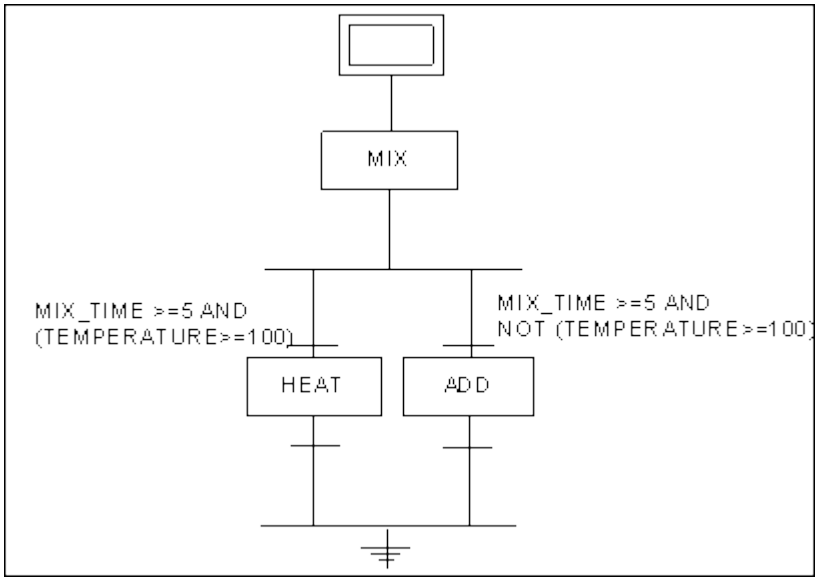
and this syntax for the other transition:

```
NOT STEPA.FAILURE
```

Using this logic, when only STEPA.FAILURE is true, the first transition executes. When only NOT STEPA.FAILURE is true, the second transition executes.

Example: NOT Operator in an Or Structure

For example, suppose you need to mix some ingredients together for 5 minutes or until the mixture is heated to 100 degrees. Depending on which condition is true first, you want the recipe to take some action, such as continuing to heat the mixture or adding more ingredients. The following figure shows the sequential function chart for this recipe.



Using NOT Operators

Defining the Order of Precedence

Batch Execution defines an order of precedence for each operator. The order of precedence determines which operators (and the values on each side of the operator) are evaluated first.

The following table lists each operator from highest to lowest precedence.

Operator Order of Precedence			
Number	**Operation**	**Operator**	**Precedence**
1	Parenthesization	(expression)	HIGHEST
2	Complement	NOT	
	Negation	- (unary minus)	
3	Multiply	*	

Operator Order of Precedence			
Number	**Operation**	**Operator**	**Precedence**
	Divide	/	
4	Add	+	
	Subtract	- (subtraction)	
5	Comparison	>, <, >=, <=	
6	Equality	=	
	Inequality	<>	
7	Boolean AND	AND	
8	Boolean OR	OR	LOWEST

Changing the Order of Precedence

You can change the order of precedence by enclosing an expression in parentheses. Parentheses have the highest precedence and all expressions within parentheses are evaluated first. Operators with the same precedence are evaluated in the order they appear from left to right.

Example: Changing the Order of Precedence

For example, consider the following expressions:

Expression	Value
6+4/2	8
(6+4)/2	5

Maintaining a Sequential Function Chart

As your needs change, you must maintain your sequential function chart to reflect conditions at your site. The following maintenance tasks are among the changes you may need to complete:

- Redefine a step.
- Delete a step.
- Reconfigure a transition.
- Delete a transition.
- Add steps.
- Add transitions.

Building Recipes

Building recipes is the process of completing the following tasks for the operations, unit procedures, and procedures required by your master recipes:

- Defining the equipment requirements.
- Implementing Active Binding.
- Creating sequential function charts.
- Completing the recipe header.
- Saving and verifying the recipe.
- Releasing the recipe to production.

For information on creating sequential function charts, refer to the Constructing a Sequential Function Chart section.

Recipe Parameters

Your main goal while building recipes is to create generic, reusable recipes. One way to accomplish

this goal is to add recipe parameters to your recipes. When you add recipe parameters, you create variables for one or more step parameters. The value of each recipe parameter is set during production, allowing you to override a step parameter and create less specific and more generic recipes. For information on recipe parameters, refer to the section Understanding Recipe Parameters and Step Parameters.

Guidelines When Building Recipes

When you are building recipes, be sure to avoid designs that could result in a deadlock situations. A deadlock occurs when a recipe is dependent upon an equipment phase in another recipe to finish.

For instance, two operations running on different units share an equipment phase. Recipe one wants shared Equipment Phase A. Recipe two wants Shared Equipment Phase B. Then, recipe one wants shared Equipment Phase B, and recipe two wants Equipment Phase A. In this scenario, the shared Equipment Phase A does not get released until B is acquired. Shared Equipment Phase B does not get release until A is acquired. A deadlock results.

Maintaining Recipes

Once the recipes are built, you need to maintain them so that they reflect your changing needs. For information on maintaining recipes, refer to the Maintaining Recipes section.

Understanding the Procedure Hierarchy

Using Batch Execution, you can create master recipes that conform to the industry-standard S88.01 Procedural model. Each level in the model is created independently and shows the steps in the level below it.

The S88.01 procedural model defines the following hierarchy of procedural levels:

Procedure – defines the strategy for accomplishing the task of making a batch. It consists of all the unit procedures defined for a recipe.

Unit Procedure – consists of operations that control the function of a single piece of equipment (unit). Multiple unit procedures can run simultaneously on different units.

Operation – consists of a series of phases. A *phase* is an individual step in the recipe. For example, MIX, HEAT, and TRANSFER are all phases that might be included in a recipe. By grouping one or more phases in sequential order, you can define operations in a recipe.

Batch Execution executes only one operation on a unit at any time. The current operation must complete before another operation can start on the same unit. Therefore, you should design operations to end where a safe hold occurs in the process.

Building a Recipe

The process of building a recipe is simple and straightforward. You begin as follows:

1. Select the recipe level you want to create, either an operation, a unit procedure, or a procedure.

   Typically, operations are built first. This approach lets you create small, reusable pieces of the overall master recipe.

2. Select the equipment requirements. This step lets you define the specific unit or unit class the recipe requires when it runs, as well as define bind types and equipment capacity. For additional information, refer to the Understanding Equipment Requirements section.
3. Define the steps in a sequential function chart. Each step represents a unit procedure, an operation, or a phase, depending on the level of the recipe you are creating. By arranging and linking the steps together, you build a recipe.

Repeat this process of building recipes for each recipe you need. As you work through this process to create operations, make them generic and reusable. You can identify which operations should be generic by examining the phases that each master recipe requires. If two or more master recipes require the same sequence of phases, consider creating a single operation for each recipe.

Similarly, if multiple master recipes require phases that differ only in the values that they set, consider creating a generic operation with recipe parameters to set the different values for each master recipe. For more information on recipe parameters, refer to the Implementation Strategies and Understanding Recipe Parameters and Step Parameters sections.

Once you create these recipes, you can arrange and link the operations together to create the unit procedures you need. Likewise, by arranging and linking unit procedures together you can create a procedure.

Using Active Binding

Active Binding allows you to dynamically bind a physical unit to a unit procedure. Batch Execution can bind and re-bind units when a batch is created, as well as during batch production.

There are three methods of binding a unit to a unit procedure:

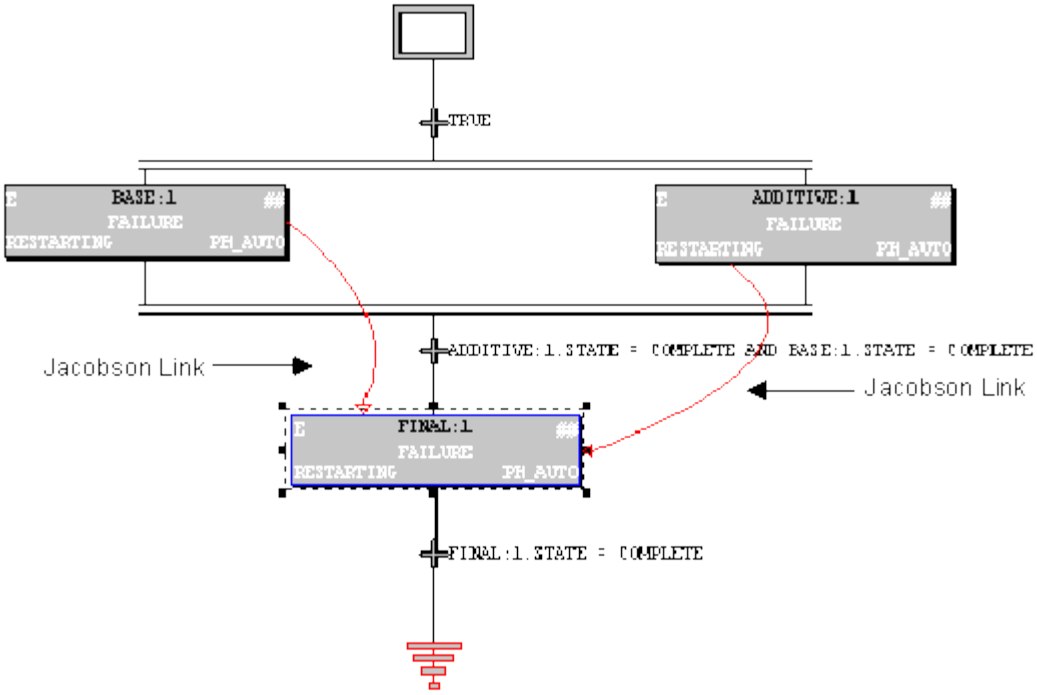
- Manually, when the recipe is created.
- During run time, either using an operator prompt or manually, if the operator has rights to do so.
- Automatically, allowing the Batch Execution Server to make the selection dynamically.

When you choose to allocate units automatically, there is certain information you must provide to the Batch Execution Server so that it can evaluate and select the correct unit on which to run a unit procedure. For example, the Batch Execution Server must know:

- Whether two unit procedures can run on the same unit or must run on different units.
- Whether the units used for two unit procedures must be physically connected.
- The size requirements of each unit procedure so that a sufficiently large unit can be selected.
- Additional equipment requirements, such as the bind type, operator rights, and scaling.

Using Jacobson Links

Frequently, two units need to be physically connected in order for a recipe to function properly. To automatically bind units to unit procedures, the Batch Execution Server must know a recipe's physical connection requirements. You specify these physical connections using Jacobson links. Jacobson links appear as red lines with a directional arrow in both the Recipe Editor and the Batch Execution Client, as shown in the following figure.



Jacobson Links in the Recipe Editor

For example, a recipe may contain a PREMIX unit procedure that transfers its contents into a reactor. There may be many reactors in the same class in the plant, but not all the reactors are physically connected to the unit on which the PREMIX unit procedure is being run. The sample toothpaste application contains a similar example. The units on which the BASE and ADDITIVE unit procedures run must both be physically connected to the unit running the FINAL unit procedure.

Forced Binding

Forced binding allows you to indicate two unit procedures that must use either the same unit or must use different units. You may have a recipe that contains two unit procedures that eventually need to be combined. It may be beneficial to force these two unit procedures to run on the same unit in order to reserve other units for less compatible unit procedures.

In a different scenario, assume that two unit procedures include potentially caustic or dangerous products, especially if any trace from the first unit procedure's ingredients mixes with ingredients from the second unit procedure. In this case, to ensure that the unit procedures use different units, use forced binding.

In the sample toothpaste application, the ADDITIVE and BASE unit procedures use forced binding to ensure that they run on different units.

Configuring a Unit Procedure's Capacity

You can define a unit procedure's capacity to indicate the amount of material the unit procedure can transfer, process, or contain. By comparing the unit procedure's capacity requirements with the defined equipment capacity of the units in the area model, the Batch Execution Server can dynamically select an appropriate unit on which to run a particular unit procedure at run time.

You set the capacity requirements at the unit procedure level in the Recipe Editor. You also have the option of overriding the settings at the procedure level for a specific step in an SFC. This allows a unit procedure to have different requirements in each procedure in which it runs.

For example, in the sample toothpaste application the MAKE_ADDITIVE unit procedure's capacity is 200 liters. However, if this same unit procedure is used in a different application that requires a much greater quantity of flavoring, the capacity requirements of the MAKE_ADDITIVE unit procedure could be changed to 500 liters.

The unit of measure used to define the equipment requirement in the Recipe Editor is validated against the UNIT_OF_MEASURE enumeration in the area model. If, for example, you select a different unit of measure in the Recipe Editor than the unit of measure defined for that unit in the Equipment Editor, this is detected during recipe verification. If at run time there is still a discrepancy in the unit of measure, the capacity logic will not be enforced.

For additional information on the UNIT_OF_MEASURE enumeration set, refer to the Equipment Configuration Manual.

Understanding Equipment Requirements

Additional equipment requirements must be defined for unit procedures and operations in order to use Active Binding. These additional equipment requirements are:

- Recipe type (class-based or instance-based)
- Bind types
- Operator rights
- Scaling

Operation-Level Equipment Requirements

At the operation level, the equipment requirements defined in the Recipe Editor are simple. You must select whether to use a class-based or an instance-based operation. In the case of a class-based operation, you specify a unit class and a default unit for the operation. If you are using an instance-based operation, you simply specify a unit on which to run the operation.

Unit Procedure-Level Equipment Requirements

At the unit procedure level, you must also define whether the recipe is class-based or instanced-based. This is done in the same manner as at the operation level. At this level, you also define capacity requirements, including whether scaling for the capacity requirement is enabled. Both of these features are described in the Configuring a Unit Procedure's Capacity section.

As part of a unit procedure's equipment requirements, you also define the:

- Bind type
- Operator rights

There are three bind types from which to choose:

Automatic Binding – this method of binding allows the Batch Execution Server to select the unit to bind to the unit procedure at run time. The Batch Execution Server makes this selection based on:

- The unit procedure capacity requirements
- Equipment pathing constraints
- Real-time conditions on the plant floor

To determine the real-time conditions on the plant floor, Batch Execution provides two Unit Status tags: Unit Ready and Unit Priority. The *Unit Ready tag* is an integer programmed in to the process controller that indicates whether a unit is available to a process or an operator. The *Unit Priority tag* is an integer that assigns a relative importance to a particular unit, as compared to other units in the same unit class. By evaluating these tags, as well as the information provided in the unit procedure capacity requirements, the Batch Execution Server can select an appropriate unit to bind to the unit procedure.

For additional information on the Unit Ready and Unit Priority tags, refer to the Equipment Configuration Manual.

Operator Prompt – this method of binding prompts the operator just before the unit procedure in question is run. The operator can then select the unit to bind to the unit procedure.

Specify at Batch Creation – this method of binding requires the operator to select a unit to bind to the unit procedure when the batch is added to the Batch List.

Along with defining the bind type, you can specify operator rights. The two types of operator rights are:

- **Modify Binding at Batch Creation** – lets operators select units to bind to unit procedures when a batch is added to the Batch List. If the operator does not select units, the batch uses the default unit bindings.
- **Modify Binding During Batch Execution** – lets operators rebind a unit to the unit procedure during batch execution.

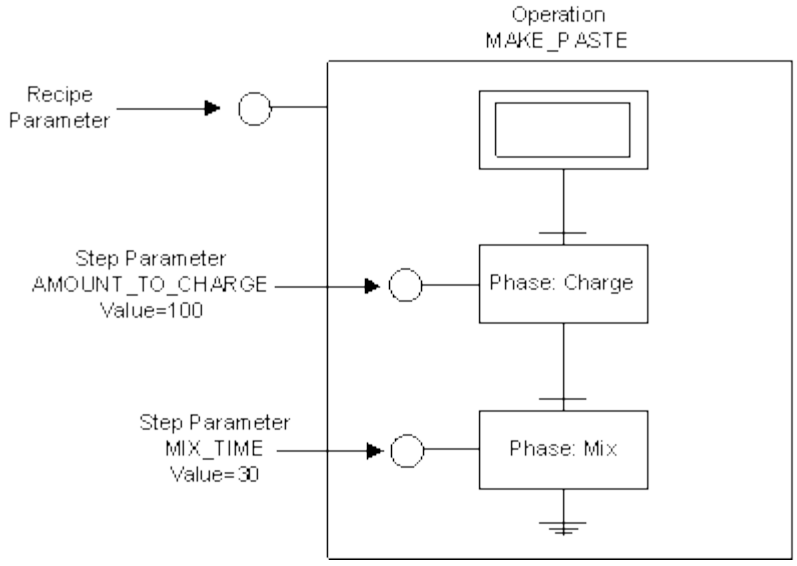
Understanding Recipe Parameters and Step Parameters

Recipe and step parameters let you create flexible and reusable recipes. Recipe and step parameters are variables that you can set to control process values such as time, temperature, and amounts.

Step parameters are defined within the entity the step represents. In the case of a unit operation, the steps represent equipment phases. At this level, step parameters are equivalent to phase parameters. In a procedure or unit procedure, step parameters represent the recipe parameters at the next level down.

Example: Using Recipe and Step Parameters

With recipe and step parameters you can pass values from one recipe level to another. For example, consider the operation in the following figure. Notice that the values for the step parameters AMOUNT_TO_CHARGE and MIX_TIME are hard coded. This operation always uses these values.



Sample Operation

You can defer the step parameter up one or more levels, which allows you to pass in any value you want at a different level. For example, to defer the step parameters AMOUNT_TO_CHARGE and MIX_TIME, you instruct the operation to disregard the hard-coded values of the step parameters (at the level you defer it at) and receive the values from another level (or levels) up from the level you specify the deferral at.

For more information on deferring parameters, refer to the section Working with Deferred Step Parameters.

Working With Recipe Parameters and Step Parameters

You add phase parameters and their default values in the area model in the Equipment Editor. When accessed from an operation in a recipe, the phase parameters are called step parameters. Each step parameter is associated with one recipe parameter. Recipe and step parameters are automatically added to your recipes. You do not need to add these parameters to your recipes.

A Recipe Parameter is automatically created for you when you defer a step parameter. At the procedure level ( ), the Recipe Parameters dialog box lists all of the deferred step parameters available at the steps contained in the unit procedure. At the unit procedure ( ) level, the Recipe Parameters dialog box lists all of the deferred step parameters in the steps at the operation level. However, at the operation ( ) level, the Recipe Parameters dialog box lists only the deferred step parameters at the phase-level.

The Recipe Parameter dialog box allows you to:

- Enter the engineering units (EGU).
- Enter the EGU High and Low information (for Real or Integers only).
- Clear or select the Scalable check box.
- Enter a Default value.
- Create a new recipe parameter with a new name and type, not associated with a step parameter.

  NOTE: *The Name and Type fields cannot be modified after you click OK.*
- Delete a recipe parameter, if it is not designated to a step parameter.

The Step Parameter dialog box allows you to:

- Select a parameter Origin option (Value, Defer, Defer to Procedure, Defer to Unit Procedure, or Operator).
- Enter the EGU High and Low information (for Real or Integers only).
- Enter a value for the parameter.
- Select the Display option check box.

Learning About Step Parameters

Each step parameter is associated with one recipe parameter. Step parameters control the same process values that recipe parameters do, but they control them at a lower level.

You define phase parameters and their default values in the area model. When accessed from an operation in a recipe, phase parameters are called step parameters. In the Recipe Editor, you have the option to do one of the following tasks with step parameters:

- Prompt the operator for a value using an operator prompt.
- Override a phase parameter's default value and set it to any value within the parameter's EGU range by using a constant value step parameter.
- Pass a step parameter value up one or more recipe levels by using the Defer, Defer to Procedure, or Defer to Unit Procedure options.

You must verify and save a recipe after making changes to a step parameter. Your changes are not propagated until you do so.

IMPORTANT: *It is very important that you verify the recipe so that your step and recipe parameters are updated. Recipes will need to be verified to the highest level that you chose to defer. For example, if you chose to Defer to Procedure, you should verify the recipes at the Procedure level so the automatic propagation of the parameters will take place. If you select to Defer to Unit Procedure, you should verify the recipe at the Unit Procedure level.*

Working with Operator Prompts

To create flexible and reusable recipes, use operator prompts. *Operator prompts* are step parameters that prompt the operator for a value prior to executing the phase. When the operator enters a value, the new value overrides the default value defined in the area model.

Example: Operator Prompts

For example, in the sample application, the phase XFER_IN uses the phase parameter AMT_TO_ADD. This phase controls the amount of ingredients to add in order to make the toothpaste base in the MAKE_BASE operation.

Now assume the operation MAKE_BASE is shared by multiple master recipes.

By using an operator prompt in the XFER_IN phase, the operator can easily adjust the phase based on the batch being produced. For example, if batches for four different types of toothpaste are being made, the AMT_TO_ADD phase parameter can be doubled or tripled, as needed. However, if only one type of toothpaste is being made, the parameter can be halved.

Working with Constant Value Options for Step Parameters

While operator prompts make recipes flexible and reusable, they also introduce the potential of operator error. An operator can incorrectly enter a number and Batch Execution has no way of knowing whether the number is too high or too low for your process. It can only detect if the value is within the defined EGU range.

You can remove this potential problem by using the constant value option for step parameters. *Constant value options for step parameters* are typically used when several master recipes share two or more operations. When you define a constant value option, you enter a specific value for it. This value lets you define the correct value prior to releasing a recipe to production. When the phase runs, the specified value automatically overrides the default defined in the area model prior to executing the phase. By automatically overriding the phase parameter's default value, you eliminate the potential of operator error.

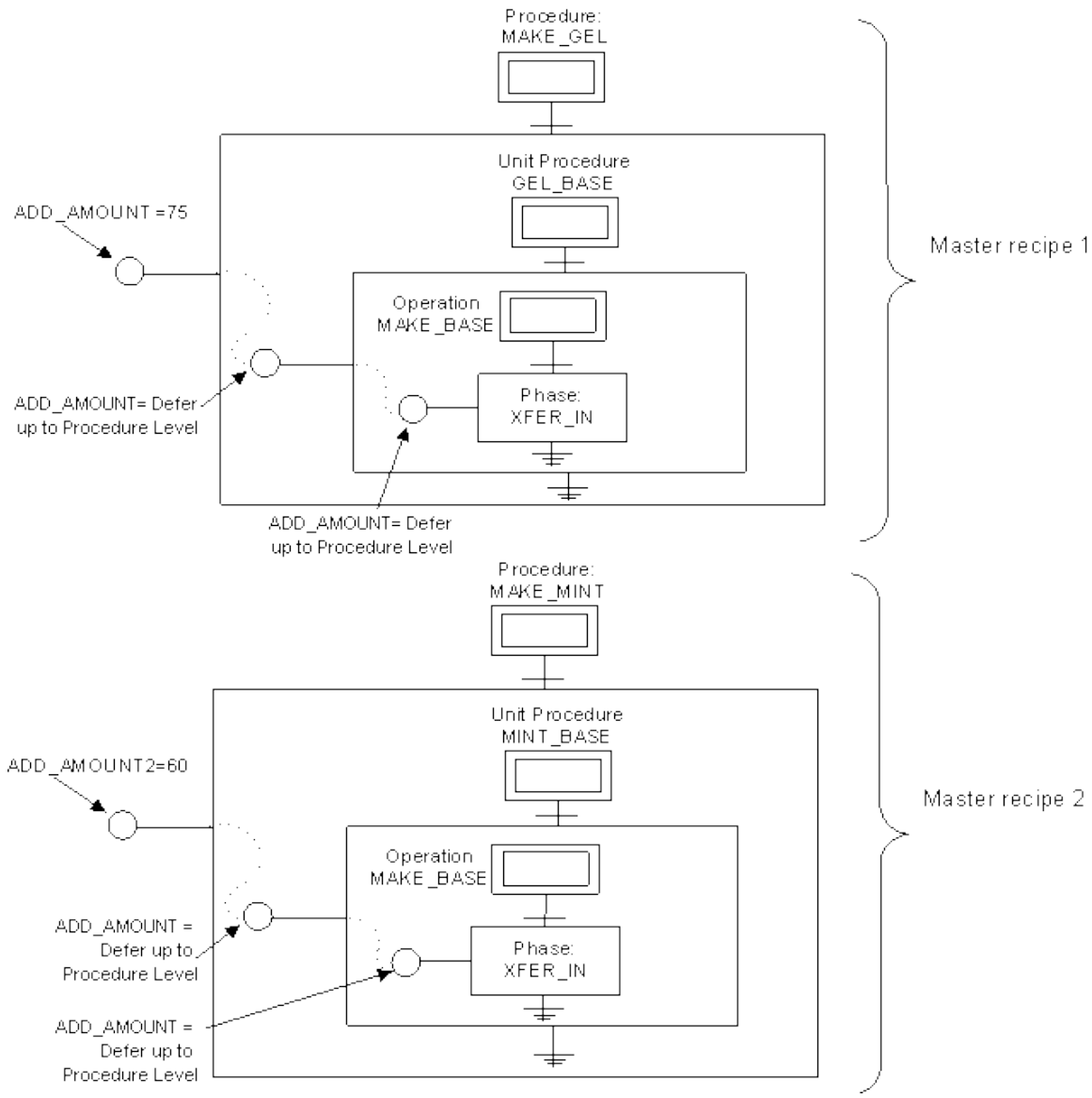
Example: Constant Value Options for Step Parameters

For example, assume the operation MAKE_BASE from the sample application is shared by two master recipes. If both master recipes require different values for the ADD_AMOUNT phase parameter, you could create two copies of the same operation, one for each master recipe and then individually define the values of each phase parameter for each master recipe.

The drawback to this approach is that if the operation changes you need to make the same changes twice, once for each copy of the operation. This can be time-consuming, particularly if the operation changes frequently.

Defer to Procedure Level

A better approach is to create one generic operation and define unique constant value options for step parameters for each master recipe. Because the values of the step parameters are unique to a master recipe, they do not affect the other master recipes. More importantly, because you define the operation only once, you save on maintenance. The following figure shows how to set up these parameters.



Using Constant Value Options for Step Parameters

NOTE: *In the above example, ADD_AMOUNT and ADD_AMOUNT2 will be available at batch creation time when you run the procedure.*

Working with Deferred Step Parameters

In addition to using constant value options for step parameters, the operations and unit procedures can use deferred parameters. A *deferred parameter* is a value that is passed up one recipe level. By deferring a step parameter, you instruct a recipe to retrieve the value from another level in a recipe, and not from the area model. The following table describes the deferral options at each level of a recipe.

Level	Deferral Options
Procedure	• Defer to Procedure • Defer
Unit Procedure	• Defer to Procedure • Defer to Unit Procedure • Defer
Operation	• Defer to Procedure • Defer to Unit Procedure • Defer

You must verify a recipe after making changes to a step parameter. Your changes are not propagated until you do so.

Example: Deferred Parameters

The figure in the Working with Constant Value Options for Step Parameters section illustrates the phase XFER_IN defined with the step parameter ADD_AMOUNT and its value is defined at the procedure level. To defer the step parameter up to the procedure level, in the Step Parameter dialog box at the phase level, select the Defer to Procedure option. This allows the automatic deferral up to the Procedure level.

When the phase is automatically deferred, the following steps occur:

1. The step parameter is deferred to the operation recipe step parameter as ADD_AMOUNT.
2. The recipe step parameter ADD_AMOUNT is in turn deferred one level up to the unit procedure recipe parameter as ADD_AMOUNT.
3. At the procedure level, the step parameters are set based on the specific recipe. At procedure level, if there is already an ADD_AMOUNT used in another operation or unit procedure, this value is incremented. For example, ADD_AMOUNT2 could be used for the step parameter at the procedure level in this scenario.

By creating this chain of parameters, you can set the value of each step/recipe parameter and pass the value all the way from the procedure-level down to the phase-level.

Modifying Recipe Parameters

A Recipe Parameter is automatically created for you when you defer a step parameter. At the procedure level (), the Recipe Parameters dialog box lists all of the deferred step parameters available at the steps contained in the unit procedure. At the unit procedure () level, the Recipe Parameters dialog box lists all of the deferred step parameters in the steps at the operation level. However, at the operation () level, the Recipe Parameters dialog box lists only the deferred step parameters at the phase-level.

In the Recipe Parameters dialog box, you can change:

- The Default Value.
- The engineering units (EGU).
- The Low EGU value (within the area model range), if the parameter is an Integer or Real data type.
- The High EGU value (within the area model range), if the parameter is an Integer or Real data type.
- Whether the parameter is scalable.
- Create a new recipe parameter with a new name and type, not associated with a step parameter.

  NOTE: *The Name and Type fields cannot be modified after you click OK.*

- Delete a recipe parameter, if it is not designated to a step parameter.

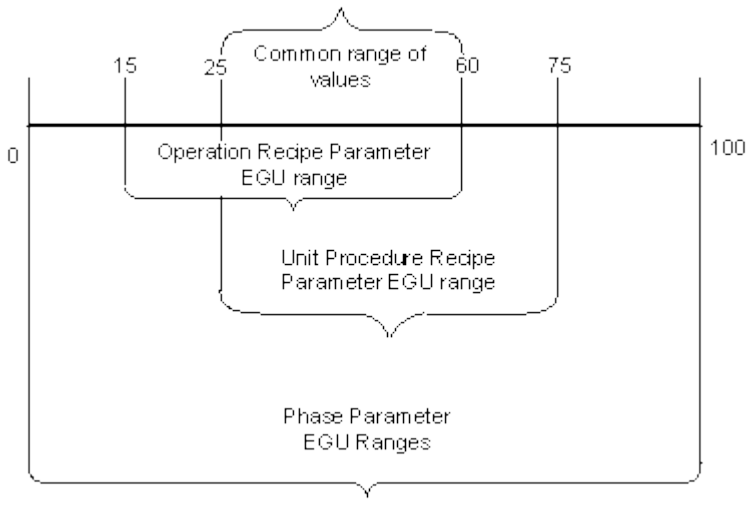
The information that you enter for the EGU range will also automatically be updated in the Step Parameter dialog box for the selected recipe step. The Step Parameter dialog box allows you to edit the EGU fields and display them in the Recipe Parameters dialog box as well. The Scalable and Default Value fields are only available in the Recipe Parameters dialog box; you cannot access these fields from the Step Parameters dialog box.

Example: Setting a Recipe Parameter's EGU Range

For example, assume you need to create an operation and a unit procedure recipe parameter for the step parameter AMT_TO_ADD. Assume the step parameter's EGU range is 0 to 100. Also assume the EGU ranges of the two recipe parameters are as follows:

Formula	**EGU Range**
Operation	15 to 60
Unit Procedure	25 to 75

The range of values that is common to the unit procedure, the operation, and the phase is 25 to 60. The following figure illustrates how this works.



EGU Ranges

The low and high level ranges for the step parameters can only be reduced from the phase parameter ranges, not expanded.

Scaling a Recipe Parameter

By scaling the value of a recipe parameter, you make the recipe more flexible and allow varied amounts of your product to be manufactured.

When you enable scaling for one or more recipe parameters and the recipe is executed, the value of each recipe parameter with the scaling option enabled is adjusted by the same amount, thereby preserving the relative proportions of the recipe.

IMPORTANT:

- *Be sure to scale a recipe parameter if the associated phase supports scaling.*
- *You cannot scale enumerations or strings.*

Finalizing a Recipe

After you create a sequential function chart, you need to do the following tasks to complete the recipe:

1. Complete the recipe header.
2. Save and verify the recipe.
3. Release the recipe to production.

Refer to the following sections for more information on each task.

Completing the Recipe Header

The recipe header contains administrative information for a recipe including the name, version number, author, version date, recipe description, and recipe batch size. Completing this information provides general information to anyone who wants to learn about and understand the recipe.

By default, when you create a recipe, the Recipe Editor supplies the following information:

- Version number
- Version date
- Procedure identifier

The Recipe Editor requires this information plus the name of the author to verify the recipe. Other information, such as the recipe description and the default batch size, is optional.

NOTE: *If you plan to scale the values in a procedure, unit procedure, or operation, make sure you complete the default batch size field in the recipe header.*

Saving a Recipe

Once the recipe header is complete, you can save the recipe.

To save a recipe, select the Save command from the File menu or click the Save button on the Recipe toolbar.

Verifying a Recipe

After saving the recipe, verify it to ensure it has been set up correctly. Verifying a recipe examines it for the following items:

- The sequential function chart must have one starting and one ending step.
- The following header fields must be completed:
    - Procedure Identifier
    - Version Number
    - Version Date
    - Author
- The equipment requirements must be valid for the specified area model.
- Each step defines valid recipe paths.
- All steps are defined, linked, named, and associated with a recipe.
- The path to an associated recipe one level down is valid.
- All recipe parameters match their definitions in the associated recipe one level up.
- All transitions are linked and have valid expressions.
- The paths to all link groups are valid.
- Units of measure are consistent between the units defined in the Equipment Editor and the unit procedure capacity requirements defined in the Recipe Editor.

If all of these items are true and the recipe verifies correctly, the Recipe Editor displays the Verification Results dialog box with a message that the recipe is complete. However, if one or more of the verification items are not true, the Verification Results dialog box appears with a list of errors found in the recipe.

Correcting Errors

If any errors are found, double-click any error in the Verification Results dialog box to display the associated recipe. Once the recipe appears:

1. Correct the problem.
2. Re-save the recipe.
3. Re-verify the recipe.

You can choose to ignore verification warnings due to inconsistent units of measure. However, when the recipe runs, Batch Execution ignores the unit procedure's equipment capacity requirements.

Releasing a Recipe to Production

Once a recipe is verified, you can release it to production. Releasing a recipe for production allows the operator to select it in the Batch Execution Client application. Until a recipe is released, it is not available to the operator.

Maintaining Recipes

As your needs change, you must maintain your recipes to reflect changing equipment and processes at your site. The following maintenance tasks are among the items you may need to do:

- Modify a recipe.
- Delete a recipe.
- Rebuild the recipe directory.
- Export a recipe as XML.

The Modifying Recipes: Overview section provides information on modifying recipes. The Deleting a Recipe section provides information on deleting recipes. For information on the recipe directory, refer to the Rebuilding the Recipe Directory section.

Modifying Recipes: Overview

Modifying a recipe is similar to creating a new recipe.

To modify a recipe:

1. Open the recipe you need to change. The recipe hierarchy appears in the Procedural View box and the recipe's sequential function chart appears in the Construction Area.
2. Modify the recipe's sequential function chart as needed.
3. Modify the recipe's parameters as needed.

4. Modify each transition in the chart as needed.
5. Modify the recipe header to reflect the changes made to the sequential function chart.
6. Modify the equipment requirements as needed.
7. Save and verify the recipe to ensure the recipe is set up correctly.
8. Release the recipe to production.

Deleting a Recipe

You can delete a recipe you no longer need using the Recipe Editor or from the desktop. Deleting a recipe with the Recipe Editor removes the file from the hard disk and removes its entry in the recipe directory. By comparison, deleting a recipe from the desktop removes the recipe from the hard disk but leaves an entry for the recipe in the recipe directory. Therefore, if you want to delete files from the desktop, you must rebuild the recipe directory immediately afterwards.

CAUTION: *Deleting a recipe permanently removes it. Do not delete a recipe unless you are sure you no longer need it.*

Rebuilding the Recipe Directory

The *recipe directory* is the master list of the available recipe files and is maintained by the Recipe Editor. When scheduling a batch, Batch Execution displays this list to the operator. Note that only recipes that have been released to production appear in the list.

In order to keep the recipe directory up-to-date, you need to rebuild it whenever you use Windows Explorer to copy a recipe file into the recipe path or delete a recipe file from the recipe path. The recipe path is defined in your project by the Batch Execution configuration file, VBEXEC.INI, on your computer.

You can rebuild the recipe directory with the Recipe Editor. When rebuilding the directory, the Recipe Editor does the following:

- Examines all the recipes in the recipe path.
- Reads the recipe header of each file.
- Writes this information to the file RECIPE.DIR, which resides in the recipe path.

After rebuilding the recipe directory, the Recipe Editor prompts you to verify all your recipes. Selecting the Yes button instructs the Recipe Editor to verify each recipe. For more information on verifying recipes, refer to the Finalizing a Recipe section.

NOTE: *The Recipe Editor automatically rebuilds the recipe directory whenever the recipe storage type for the project does not match the storage type of recipes listed in the recipe directory.*

Exporting a Recipe

In the Recipe Editor, you can export a recipe step as an XML file. Only the step (procedure, unit procedure, or operation) that is currently displayed in the Recipe Editor is exported. Steps (unit procedures or operations) that appear underneath the selected step are not included in the export. This allows you to easily compare XML versions of the recipe step.

After you save a recipe as an XML file, you can use any XML viewer to review it. You can also use XML DIFF tools to compare it with previous XML files exported. The XML Schema Definition file for the XML file you export, BatchRecipe.xsd, is stored in the Batch Bin folder, which by default is C:\Program Files\Proficy\Proficy Batch Execution\BIN.

Advanced Topics

The sections that follow describe how to create phase link groups and class-based recipes. Both tasks require specific setups in your area model. If your area model is not set up to take advantage of these features, refer to the Equipment Configuration Manual to learn how to set up the area model.

These sections also explain how to save your recipes to a relational database. This feature requires you to set up a relational database and specify the data source in your project. For more information on each task, refer to the System Configuration Manual.

Synchronizing Phases

Depending on your needs, you may want one phase to communicate with another phase. Typically, this happens when you need to synchronize the actions of both phases.

You can synchronize two or more phases by creating a phase link group. A *phase link group* is a list of phases that communicate with each other. Each phase can belong to different operations but they must reside in the same procedure. Each phase in the link group must have the same number of message partners.

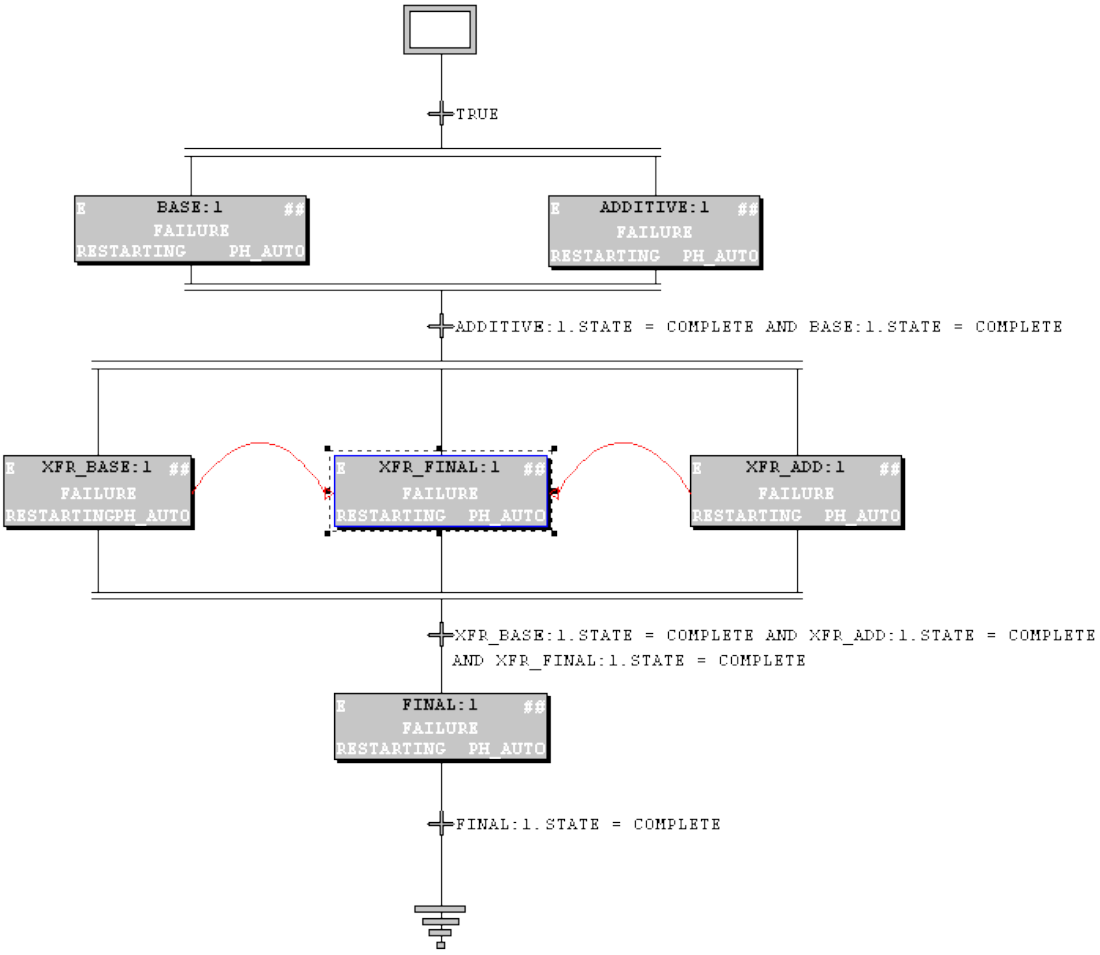
In order for synchronized phases to run, all phases must be active simultaneously. The first phase in the group sends a request to the Batch Execution Server. The Batch Execution Server does not clear this request until the other phases in the phase link group send requests to the Batch Execution Server, too. The Batch Execution Server then clears all requests simultaneously.

For more information on linking units and defining message partners and requests, refer to the Equipment Configuration Manual.

Example: Synchronizing Phases

For example, in the Demo sample application, shown in the following figure, the unit procedures XFR_BASE and XFR_ADD each have phases called XFR_OUT_LINK. These phases transfer material to the unit procedure XFR_FINAL, which has a phase called XFR_IN_LINK. In order for the transfer to be successful, these three phases must be synchronized with each other. The valves in the two XFR_OUT_LINK phases must be opened to transfer the material, and the valve in the XFR_IN_LINK must be opened to accept the material.

Now assume one of the XFR_OUT_LINK phases becomes active before the second XFR_OUT_LINK phase. The first XFR_OUT_LINK phase does not start because the other phases in the link group are inactive. As a result, the XFR_OUT_LINK phase waits for the two other phases in its link group to become active. Once all the phases are active, they synchronize and start running.



Synchronizing Phases

Building Class-Based Recipes

When you create an operation, you can assign a specific unit to it. This setup is adequate if the operation only runs on one unit. However, when an operation needs to run on multiple units, defining a specific unit makes the operation inflexible and forces you to create an operation for each unit.

A better approach is using class-based recipes. A *class-based recipe* is a recipe that defines equipment in terms of a unit class and not specific unit instances. As a result, the recipe can run on multiple units and can be assigned to any unit in the unit class at run time.

For more information on using class-based recipes as part of an implementation strategy, refer to the Implementation Strategies section.

Class-Based Recipe Prerequisites

To create a class-based recipe, your area model must define class-based units. The units can have different phases; however, the only phases available to use in the class-based recipe are the phases that are identical on all units in the class.

For example, two mixers, MIX1 and MIX2, have several phases in common: a Transfer-In phase, an Agitate phase, and a Transfer-Out phase. However, MIX2 also has a Heat phase. As long as only the Transfer-In, Agitate, and Transfer-Out phases are used in a recipe, the recipe can be class-based. If the recipe calls for the Heat phase, these units can no longer be used in the class-based recipe.

For more information on setting up class-based units, refer to the Equipment Configuration Manual.

Creating a Class-Based Recipe

In general, creating a class-based recipe is similar to creating an instance-based recipe. The only difference is that instead of selecting a specific unit you select a unit class for the recipe's equipment requirements.

Building Transitions for Class-Based Recipes

When you create a class-based recipe, you can build transitions as described in the Constructing a Sequential Function Chart section. The syntax for transitions using units and tags is slightly different than described because the recipe is class-based.

IMPORTANT: *When building class-based recipes, do not reference equipment instances in the initial transition. If you do so, the initial transition will be dependent upon the equipment being acquired before it can run, and if it is not available, an error occurs.*

Syntax: Class-Based Units and Tags

The syntax is as follows:

```
unitclassname.tagclassname
unitclassname.tagclassname operator value
```

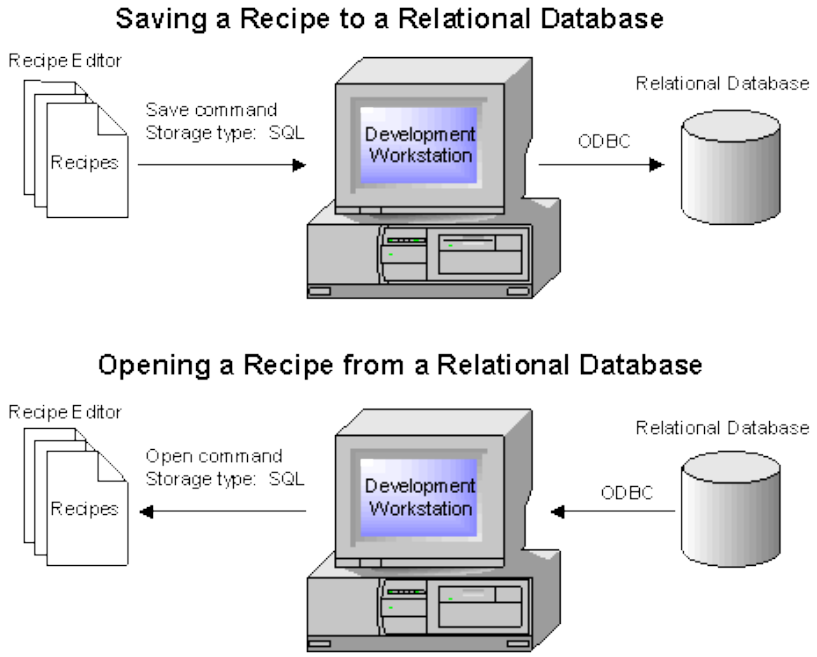
For more information about units and tags in transitions, refer to the Constructing a Sequential Function Chart section.

Saving Recipes to a Relational Database

Depending on how your project is set up, Batch Execution saves all the recipes in the project to files on your hard disk or to a relational database. When the project is set up to save recipes to a relational database, the Recipe Editor saves the recipe locally and then converts the recipe for storage in the relational database.

When you subsequently open the recipe, Batch Execution reads the recipe from the relational database and converts it back into its original format. Then the Recipe Editor opens the file and displays it.

The following figure shows both processes.



Saving and Opening SQL-Based Recipes

For information on configuring your relational database, refer to the Custom Applications manual.

Converting Recipes

In addition to saving recipes to a relational database, you can also convert all the recipes in the project from either format supported by the Recipe Editor:

- File-based recipes
- SQL-based recipes

Implementation Strategies

Before building your recipes, take some time to think about each recipe and how best to represent the recipe with Batch Execution. The sections that follow discuss the issues you need to consider as you plan your recipes and provides suggestions for taking advantage of Batch Execution:

- Design Goals
- Design Strategies for Recipes
- Design Strategies for Sequential Function Charts

Design Goals

Your primary design goal when developing recipes is to create small, flexible operations that can be reused and recombined in different ways. This approach provides you with the flexibility and power available in Batch Execution. For example, instead of creating five custom operations for five different mixers, consider designing one generic operation that can be used by the five mixers.

To achieve this goal, you need to plan your recipes and decide which phases are incorporated into each operation. However, as a recipe designer, you are tied to the flexibility that was designed into the area model. Therefore, if the area model is being developed concurrently with your recipes, you may want to consult with the engineer developing the area model to ensure the database is set up for your needs.

Design Methodologies

Recipes are best designed with a "bottom up" approach; design small, flexible operations first, followed by unit procedures and, ultimately, procedures. This is a fundamentally different approach than designing an area model. To design an effective area model, an engineer should employ a "top down" approach. This type of design strategy encourages the engineer to evaluate the "big picture," and therefore develop the process cells first, followed by the units and then the equipment phases.

Design Strategies for Recipes

You can use the following strategies to create small, flexible operations:

- Use recipe parameters to create generic operations for common tasks.
- Create class-based recipes when an operation can run on two or more pieces of equipment.
- Use Active Binding to dynamically allocate an appropriate, available unit at run time.

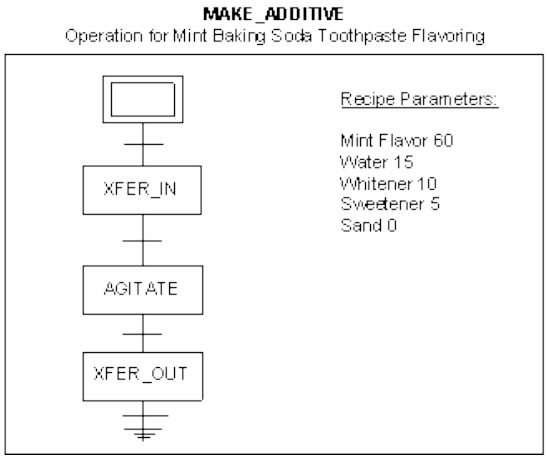
Using Recipe Parameters

One important aspect of recipe design is balancing the complexity of each operation against its flexibility. An operation may have hard-coded process values defined in the individual phases that comprise the operation. Such operations are simple to use and set up, but not very flexible.

A more flexible approach is to add a recipe parameter to the operation. A *recipe parameter* is a variable that represents a process value. By including a recipe parameter in an operation, you make it generic and reusable because the recipe parameter acts as a placeholder for specific values. Use recipe parameters to override the hard-coded value in a phase and let you specify any value you need, depending on the batch in production. Once a value is specified, the value is passed to a phase in the area model. For additional information, refer to the Building Recipes section.

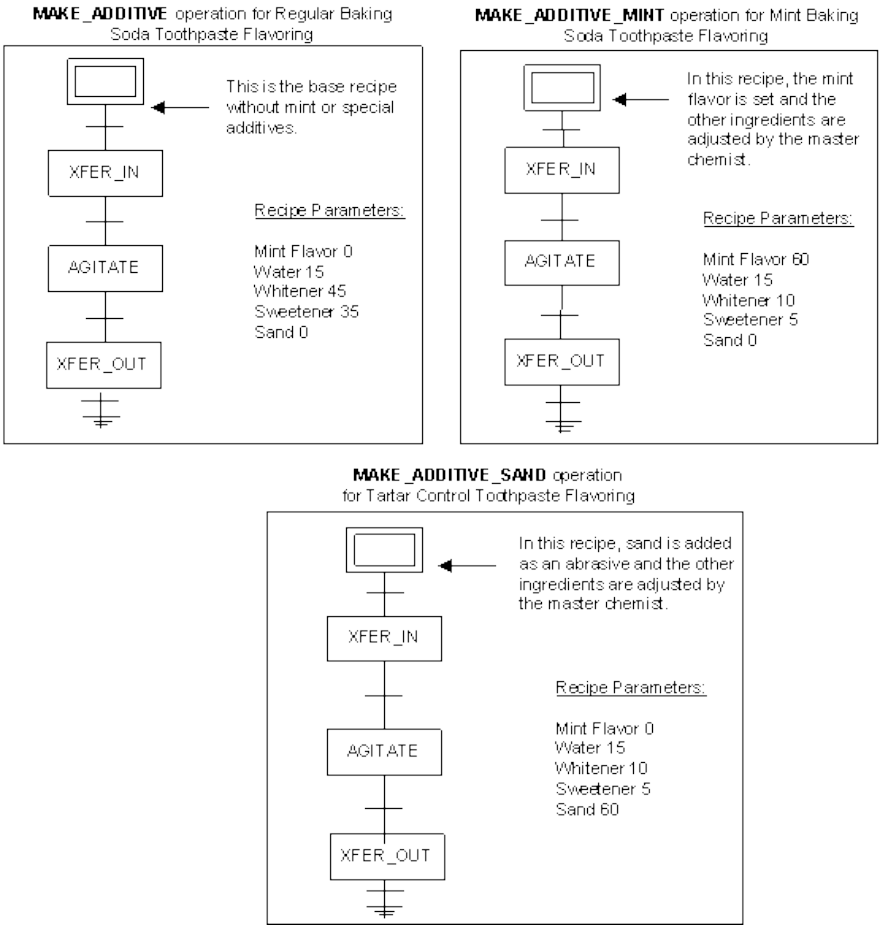
Example: Using Recipe Parameters

For example, the sample application uses an operation, MAKE_ADDITIVE, to create the toothpaste flavoring. This operation mixes ingredients to create the flavoring. To make this recipe more generic, the ingredient amounts could be defined in recipe parameters, as shown in the following figure.



Sample Recipe Parameter

Once the recipe parameter is defined, you can specify any value you need. By specifying different values for the ingredient amounts, you can make many different products from this operation, as shown in the following figure.



Using Recipe Parameters

Using this approach makes the operation more flexible but also adds complexity. In order for each recipe parameter to work as intended, you must associate it with a step parameter in the recipe. By making this association, you allow the recipe parameter to pass its value down to the step parameter, which in turn passes this value to the area model and overrides the hard-coded values in each phase.

Creating Recipes That Can Run on Different Units

Recipe parameters make operations generic so that they can adjust a process value. However, you can also make operations generic so that they run on multiple units of the same type and not only on a specific unit.

Operations that can run on multiple units of the same type are called *class-based recipes*. Class-based recipes let you select a unit class and not a specific unit. The *unit class* defines the common attributes for the type of equipment you are working with. Unit classes define the:

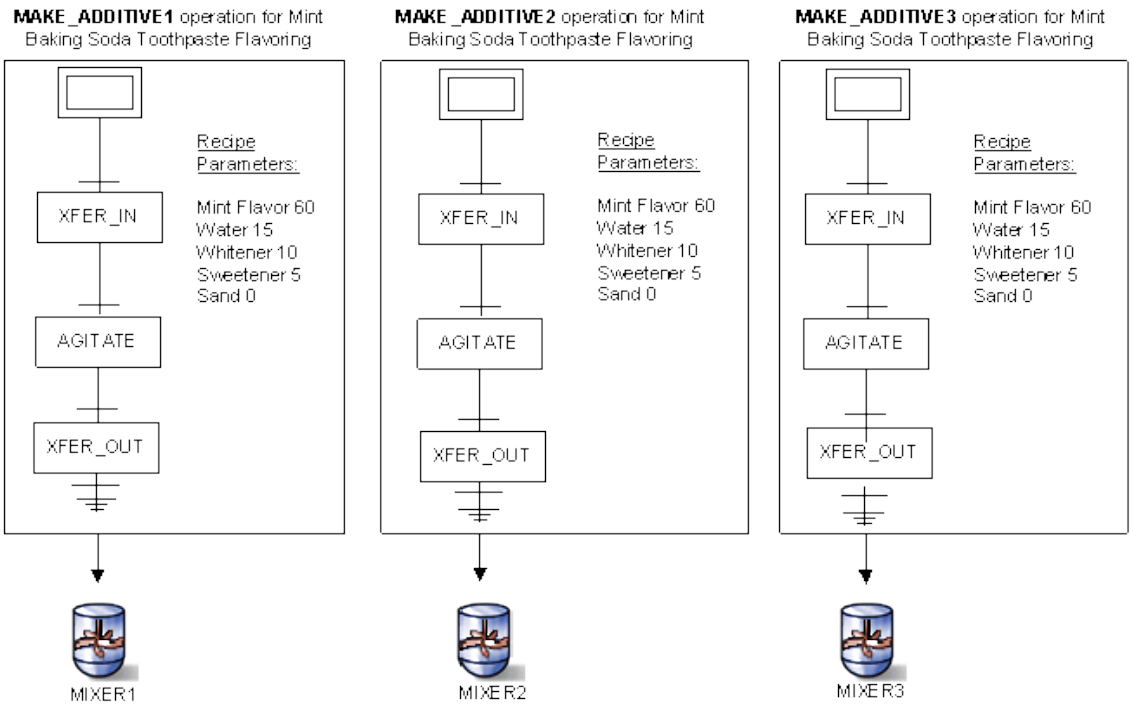
- Phases on all units in the class.
- Tags on all units in the class.

This common information is shared with each specific unit, or *instance,* in the class. Each instance defines unit-specific information, such as the unit's ID, for communicating with the process hardware.

The unit class also acts as a placeholder and template for a specific unit. As a result, when the procedure containing the class-based operation is scheduled, either the operator is prompted to select a unit instance from the unit class specified for the operation or the Batch Execution Server will select a specific unit. This allows Batch Execution to bind a specific unit to the operation and begin production.

Example: Creating Class-Based Recipes

For example, suppose you want to include the MAKE_ADDITIVE operation shown in the following figure in three different procedures running on three different production lines.



Running Recipes on Similar Units

If the operation is instance-based, you need to create three operations, one for each mixer, as shown in the preceding figure. Notice that aside from the equipment defined, the operations are identical.

To avoid this situation and simplify your development effort, make the operation class-based. By creating a class-based recipe, you enable the operation to run on any of the three mixers because a unit class is defined instead of a unit instance.

This approach allows you to create one generic operation instead of three specific ones, saving you time and maintenance. By implementing active binding, operators or the Batch Execution Server can make run-time decisions that assign any appropriate mixer in the unit class to the operation, providing flexibility.

For more information on creating class-based recipes, refer to the Advanced Topics section.

Design Strategies for Sequential Function Charts

In one sense, a recipe is a series of steps (phases) that execute in a specific order. With Batch Execution, you can create one large operation or you can break the recipe into small, manageable chunks and then arrange the chunks according to the procedural hierarchy.

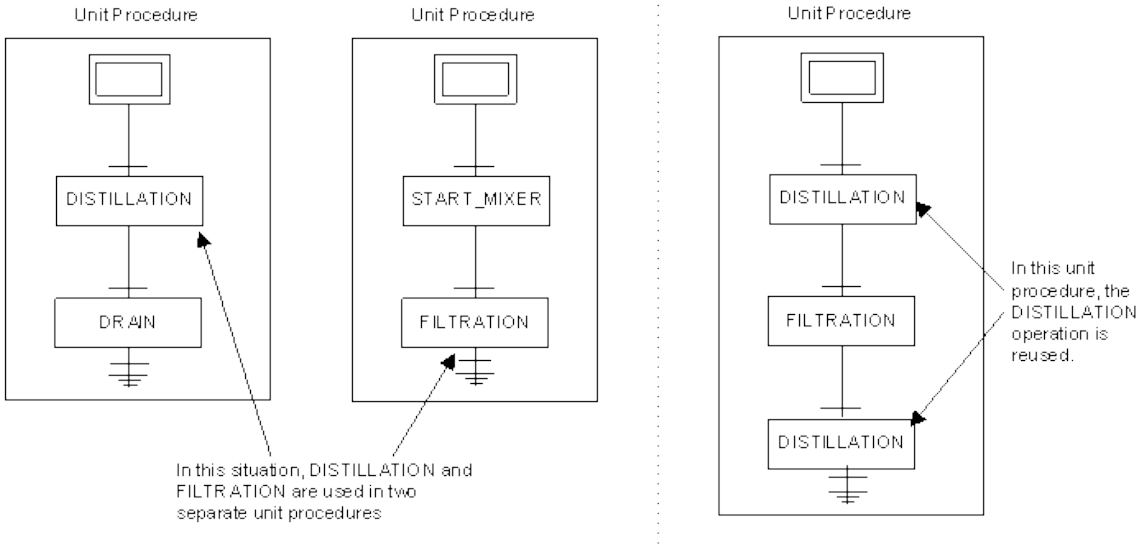
Typically, the latter is done because:

- Smaller recipe parts are easier to create and maintain.
- Arranging parts of a recipe according to the procedural hierarchy lets other process engineers understand the recipe's logic.
- Creating procedures and unit procedures lets you coordinate the function of different operations easily.

Of course, if you have a recipe that is self-contained and does not require interaction with other recipes, for example a clean-in-place operation, you can create and run the operation by itself. However, you will probably need to break more complex master recipes into smaller fragments.

When you divide the recipe into parts, you need to consider which phases are included in the same operation. For example, suppose you have a FILTRATION operation that continuously mixes ingredients to create a mixture with a specific consistency and a DISTILLATION operation that heats a mixture to a specific temperature.

Should these operations be combined or should they be separate operations? The answer depends on your needs. For example, you may want to reuse the operations in several different unit procedures, as the following figure shows.



Creating Small, Separate Operations

However, you may only need to control the temperature while mixing some ingredients. In that case, you could build the temperature control into the FILTRATION operation, as the previous figure illustrates.

Formulations Management

A master recipe contains the instructions to generate a product. These instructions identify the equipment necessary to generate the product, the coordination of the equipment as the product is generated, and parameters defining ingredient amounts, equipment settings, and process times involved in generating a batch of product.

Changing the instructions in the master recipe can have an enormous impact on end-user product quality and cost. Even if you only make a few small changes, master recipe changes often involve expensive revalidation of recipes.

The purpose of formulations management is to allow recipe parameters to be edited separately from the master recipe. Batch still supports the product quality features necessary for tracking and managing changes in the master recipes and formulations; however, formulation changes do not occur within the master recipe.

Formulations management optimizes recipe reuse. You can create multiple formulations for the master recipe. With formulations management, you only update the formulation parameter value; you do not have to reverify the entire recipe each time you want to make a change, which also leads to less cost. For example, you can test a product with different parameters (a different temperature, a different material) without requiring a revalidation of the entire master recipe. You can reuse master recipes for different purposes. For example, you can have the same toothpaste recipe, but different flavors that you want to produce using different formulations. Formulations management allows you to do this.

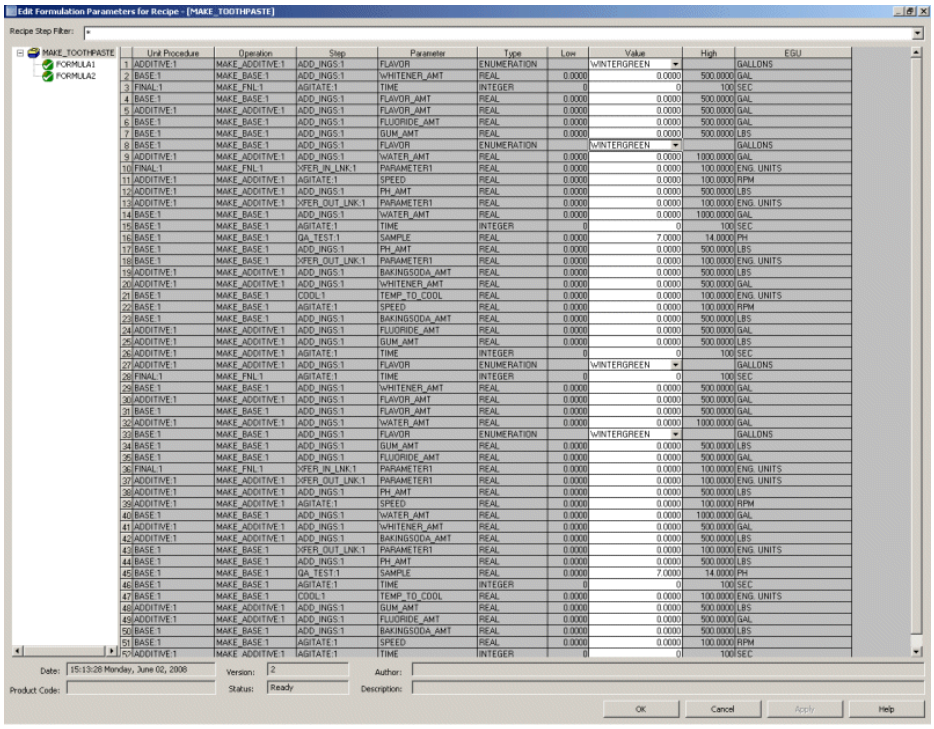
Additionally, you can define global parameters that do not change. The parameters apply to the entire project. For instance, you may always want each recipe to use the same temperature parameter. You can define this parameter at global level within the Edit Formulations Parameters dialog box. When you do this, the parameter is not available for editing in a product formulation.

In Batch Execution, formulations and global parameters are saved to a text file in XML format. They have a .FMX file extension and are saved to the project's Recipe folder. For example, for the DEMO project, if you installed to the default location, this folder is C:\Program Files\Proficy\Proficy Batch Execution\Projects\DEMO\RECIPES.

Unlike recipes, formulations currently cannot be printed.

Creating a Product Formulation or Global Parameter

To create a product formulation or global parameter, you use the Recipe Editor. Open a recipe, and on the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters for Recipe dialog box appears. An example of this dialog box with two formulations already created is illustrated in the following graphic.



The tree view area displays the selected recipe, plus its associated formulations and global parameter. Only one global parameter set per project is allowed.

The main area of the Edit Formulation Parameters dialog box displays a list of parameters for the selected formulation or global parameter set. The Recipe Step Filter drop-down list allows you to filter the list of procedures that appear in the list. You can click a column heading to sort the items in the list. Click the column header again to sort in the opposite direction.

To add a product formulation or global parameter set for the selected recipe, right-click the recipe in the tree view, and select Add Product Formulation or Add Global Parameter Set.

Formulation Status

Each formulation allows you to assign a status to it, to manage formulation development, similar to recipe development. The available formulation statuses include: **Draft**, **Ready**, **Approved**, or **Withdrawn**.

When you create or edit a formulation, the values that appear in the Status drop-down list in the Header dialog box change depending upon the last assigned value. For example, for a new formulation, you will only see Draft or Ready appear in this drop-down list. For a Ready formulation, Ready, Draft, and Approved appears in this list. For an Approved recipe, Approved, Ready, and Withdrawn appears in this list, and so on.

For electronic signatures and iFIX, if you have the "Change Formulation Status" privilege, you can change the status from one state to another state.

The following table describes these states in more detail.

Status	**Description**
Draft	The formulation is under development and not ready to be run in a production environment. A Draft status formulation cannot be scheduled to execute on the Batch Sever. All formulations, by default, are Draft status.
Ready	The formulation is complete, but not yet approved for regular production. You might want to schedule a Ready status formulation to execute a test run, but in general, a Ready batch is not run.
Approved	The formulation is complete, and approved for regular production.
Withdrawn	The formulation is no longer valid. You cannot schedule this formulation for running on the Batch Server.

Running Formulations

After you create your formulations, in order to run a formulation, the following must first be true:

- The formulation must be in a Ready or Approved state.
- The master recipe must be verified and released to production.
- Formulation must be valid (the area model should not be modified since the master recipe and the formulation were last verified).

You can run your formulations from the following Batch Execution applications:

- Batch Client
- BatchAdd ActiveX Control

Optionally, you can also run your formulations from Visual Basic. For an example of how to schedule a formulation, see VBISBatchControl5.Add (Unit Binding Only). For an example of how to run a formulation see VBISBatchControl5.Command.

NOTE: *To use the example in VBISBatchControl5.Command to run a formulation, change the line BatchCommand = "REMOVE" to BatchCommand = "START".*

You schedule a formulation just as you would a recipe. For information on running recipes, refer to the Operations Manual.

Formulations Dialog Boxes

The Batch Execution application includes the following dialog boxes for formulations management (listed in alphabetical order):

- Edit Formulation Parameters for Recipe Dialog Box
- Header (Formulations) Dialog Box
- Header (Global Parameter Sets) Dialog Box

Edit Formulation Parameters Dialog Box

The Edit Formulation Parameters dialog box displays the following items:

Tree Area

Displays the recipe procedure, unit procedure, or operation and its associated formulations and global parameter sets in tree view.

Date

Displays the date the selected formulation or global parameter set last changed. For example, a valid date might be: 11:54:31 Monday, June 23, 2008.

Version

Displays the internal version number of the selected formulation or global parameter set.

Author

Displays the name of the person who created this selected formulation or global parameter set.

Product Code

Displays the product code for the selected formulation.

Status

Displays the status of the selected formulation.

Description

Displays the description of the selected formulation.

Header (Formulations) Dialog Box

The Header (Formulations) dialog box displays the following items:

Name

Enter a name for the formulation. This name must be unique to the project. You cannot have a duplicate formulation name within the entire project.

Version

Displays the internal version number of the formulation. Each time you save the formulation with a change, this value increments.

Version Date

Displays the date the formulation last changed. For example, a valid date might be: 11:54:31 Monday, June 23, 2008.

Author

Enter the name of the person who created this formulation.

Product Code

Optionally, enter the user-defined code assigned to this formulation.

Description

Optionally, enter a description for this formulation.

Status

Select a status for the formulation: Draft, Ready, Approved, or Withdrawn. The values that appear in the drop-down list change depending upon the last value.

For example, for a new formulation, you will only see Draft or Ready in this list. For a Ready formulation, Ready, Draft, and Approved appears in this list. For an Approved recipe, Approved, Ready, and Withdrawn appears in this list.

Header (Global Parameter Sets) Dialog Box

The Header (Global Parameter Sets) dialog box displays the following items:

Name

Displays the read-only name, “Global_Parameters."

Version

Displays the internal version number of the global parameter set. Each time you save the formulation with a change, this value increments.

Version Date

Displays the date the global parameter set last changed. For example, a valid date might be: 11:54:31 Monday, June 23, 2008.

Author

Enter the name of the person who created this global parameter set.

How Do I...

The following sections provide steps on how to use formulations management:

- work with formulations
- work with parameters
- sort product formulations and global parameter sets
- filter product formulations and global parameter sets

Working with Global Parameters

The following sections provide steps to help you work with formulation parameters:

- Creating a Global Parameter Set
- Deleting a Global Parameter Set
- Editing a Global Parameter Set
- Making a Parameter Global

Creating a Global Parameter Set

To create a global parameter set:

IMPORTANT: *Your recipe should be validated and saved before you perform these steps.*

1. In the Recipe Editor, open or create a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, right-click the recipe name, select New, and then click Global Parameter Set. The Header dialog box appears.

   NOTE: *You can create only one global parameter set per recipe. The New Global Parameter Set option is unavailable if a global parameter set already exists.*

4. In the Author field, enter a name.
5. Click OK.

Deleting a Global Parameter Set

To delete a global parameter set:

1. In the Recipe Editor, open a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, right-click the Global Parameter set that you want to delete, and select Delete Global Parameter Set. A message box appears asking if you want to complete the action.
4. Click Yes to continue.

Editing a Global Parameter Set

To edit a global parameter set:

1. In the Recipe Editor, open a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, right-click the Global Parameter Set you want to edit, and select Edit Header. The Header dialog box appears.
4. In the Author field, edit the name.
5. Click OK.

Making a Parameter Global

To make a parameter global:

1. In the Recipe Editor, open or create a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, select the Global Parameter Set that contains the parameter you want to make global.
4. In the Global column, in the row of the parameter you want to make global, click the check box. You may need to horizontally scroll to view the Global column. The Value field becomes available for you to edit. The next time you open a product formulation, this parameter (now global) will not appear in the list since it is now in the global parameter set.

Working with Product Formulations

The following sections provide steps to help you work with formulations:

- Creating a Formulation
- Copying a Formulation
- Deleting a Formulation
- Editing a Formulation
- Updating a Formulation
- Updating All Formulations

Creating a Formulation

To create a formulation:

IMPORTANT: *Your recipe should be validated before you perform these steps.*

1. In the Recipe Editor, open or create a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, right-click the recipe name, select New, and then click Product Formulation. The Header dialog box appears.
4. In the Name field, enter a name that is unique throughout the whole project.
5. In the Status field, select the status of the new formulation: Draft or Ready. Keep in mind that you can only run Ready and Approved formulations.
6. Click OK.

Copying a Formulation

To copy a formulation:

1. In the Recipe Editor, open or create a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, right-click the product formulation name that you want to copy, click Copy to New Formulation. The Header dialog box appears.
4. In the Name field, enter a name that is unique throughout the whole project.
5. In the Status field, select the status of the new formulation: Draft or Ready. Keep in mind that you can only run Ready and Approved formulations.
6. Click OK.

Deleting a Formulation

To delete a formulation:

1. In the Recipe Editor, open a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, right-click the product formulation name you want to delete, and then click Delete Formulation. A message box appears.
4. Click Yes to continue and delete the recipe.

Editing a Formulation

To edit a formulation:

IMPORTANT: *Your recipe should be validated and saved before you perform these steps.*

1. In the Recipe Editor, open a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, right-click the product formulation name you want to edit, and select Edit Header. The Header Dialog Box appears.
4. In the Name field, edit the name.
5. In the Author field, edit the name.
6. In the Product Code field, edit the code.
7. In the Description field, edit the description.
8. In the Status field, select a new status (Draft, Ready, Approved, or Withdrawn).
9. Click OK.

Updating a Formulation

To update a formulation:

1. In the Recipe Editor, open a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, right-click the formulation that needs to be updated, and click Update Formulation.

   NOTES:

   - *When you update an invalid formulation, Batch checks whether each parameter exists and whether it is within range. It does not validate the recipe. You must validate the recipe separately.*
   - *The Update Formulation command is not available on the right-click menu for valid formulations (and when green check marks appear next to all formulations).*
   - *An invalid formulation (shown with a red exclamation point in the tree view of the Edit Formulation Parameters dialog box) can show that a parameter has been added (the parameter value is set to default value), a parameter that has been removed, or a parameter has a value outside of the valid range (the parameter value is set to default value). An "Action" column displays the invalid information. Clicking Update Formulation automatically applies the changes to the formulation – the parameter will be removed from the formulation, the added parameter will be added to the formulation, any parameters that had the invalid range error will be updated with the current value of the parameter, and the Action column will be removed.*

Updating All Formulations

To update all formulations:

NOTE: *When you update invalid formulations, Batch checks if the parameter exists and if it is within range. It does not validate the recipe. You must validate the recipe separately. The Update Formulation command is not available in the right-click menu for valid formulations (and when green check marks appear next to all formulations).*

1. In the Recipe Editor, open a recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. In the tree view, right-click the recipe (at the top of the tree) that needs to be updated, and click Update Formulation.

Sorting Product Formulations and Global Parameter Sets

To sort product formulations and global parameter sets:

1. In the Recipe Editor, in the tree view area, select a procedure, unit procedure, or operation recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. Select a product formulation or global parameter set.
4. Click a column header to sort the data. Click the column header again, to sort in the opposite direction.

Filtering Product Formulations and Global Parameter Sets

To filter product formulations and global parameter sets:

1. In the Recipe Editor, in the tree view area, select a procedure, unit procedure, or operation recipe.
2. On the Recipe menu, click Formulation Parameters. The Edit Formulation Parameters dialog box appears.
3. Select a product formulation or global parameter set.
4. In the Recipe Step Filter drop-down list, select a filter. Click * to display all recipes. The items in the list are filtered by your selected options.

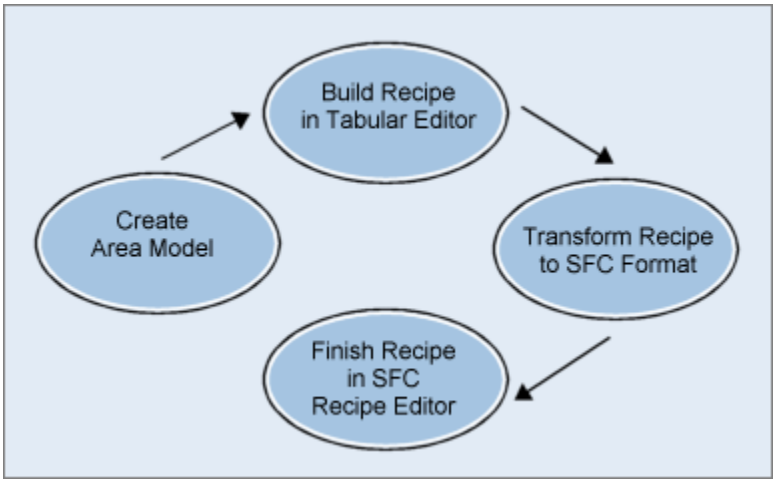
Understanding the Tabular Recipe Editor

The Tabular Recipe Editor is provided as an alternative to the Batch Execution Recipe Editor. Instead of the SFC view used in the Batch Execution Recipe Editor, the Tabular Recipe Editor allows for a linear view of recipe creation. If you have simple recipes that you modify a lot, you may find the Tabular Recipe Editor's spreadsheet view useful for recipe development. The tab delimited exports from the Tabular Recipe Editor allow for use of other spreadsheet programs such as Microsoft Excel in your recipe development.

The Tabular Recipe Editor is also helpful if you need to quickly generate new recipes from existing ones. All you need to do is open a tabular recipe, make a few quick changes, and save it under a new name. You can quickly generate the SFC from the Tabular Recipe Editor, so that you don't need to tediously create one from scratch.

The Tabular Recipe Editor does not change the fact that you need an area model and an SFC recipe to run a batch in the Client. To use the Tabular Recipe Editor you first need to create your area model. Then, you can build your recipe in the Tabular Editor, and then transform the recipe to the SFC format. And, finally, you can open the SFC format in the main Recipe Editor and validate it. The following figure illustrates how this process works.

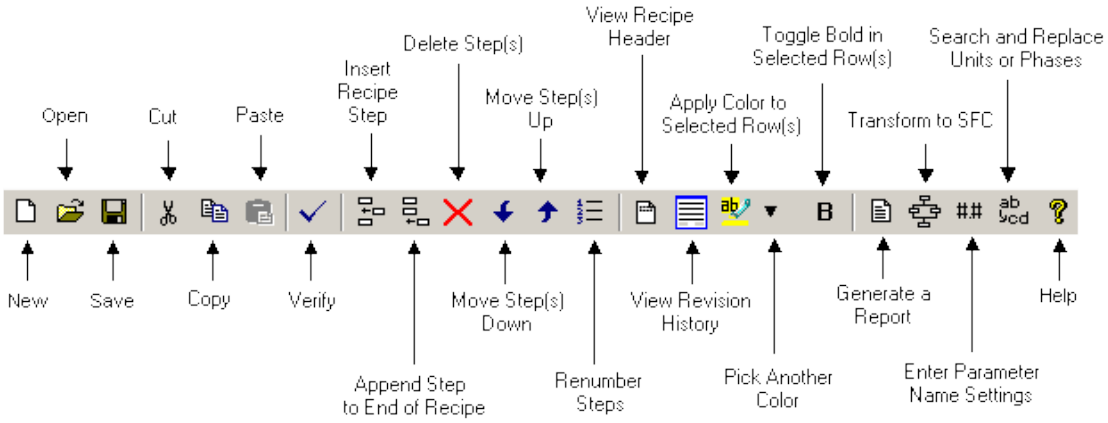
NOTE: *Recipe transformation is one-way. Once a recipe is transformed to SFC format, it cannot be transformed backwards to the Tabular Recipe Editor format.*



A maximum of 2000 rows of steps can be created, per recipe, in the Tabular Recipe Editor.

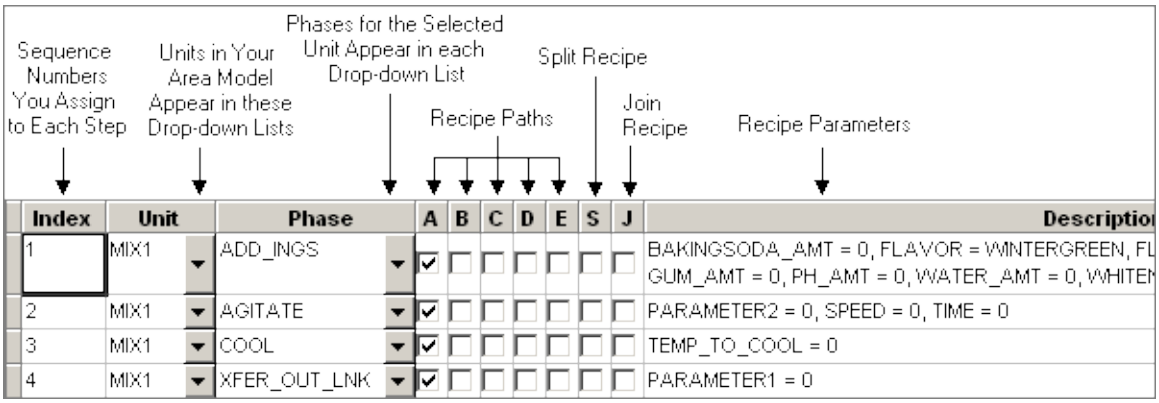
Using the Tabular Recipe Editor Toolbar

Across the top of the Tabular Recipe Editor window is the Recipe toolbar. This toolbar, shown in the following figure, provides quick access to common recipe operations such as inserting or moving recipe steps, formatting the recipe spreadsheet, saving a recipe, verifying a recipe, and displaying phase parameters.



Understanding the Grid in the Tabular Recipe Editor

The following graphic illustrates the top half of the recipe steps spreadsheet and what displays in it.

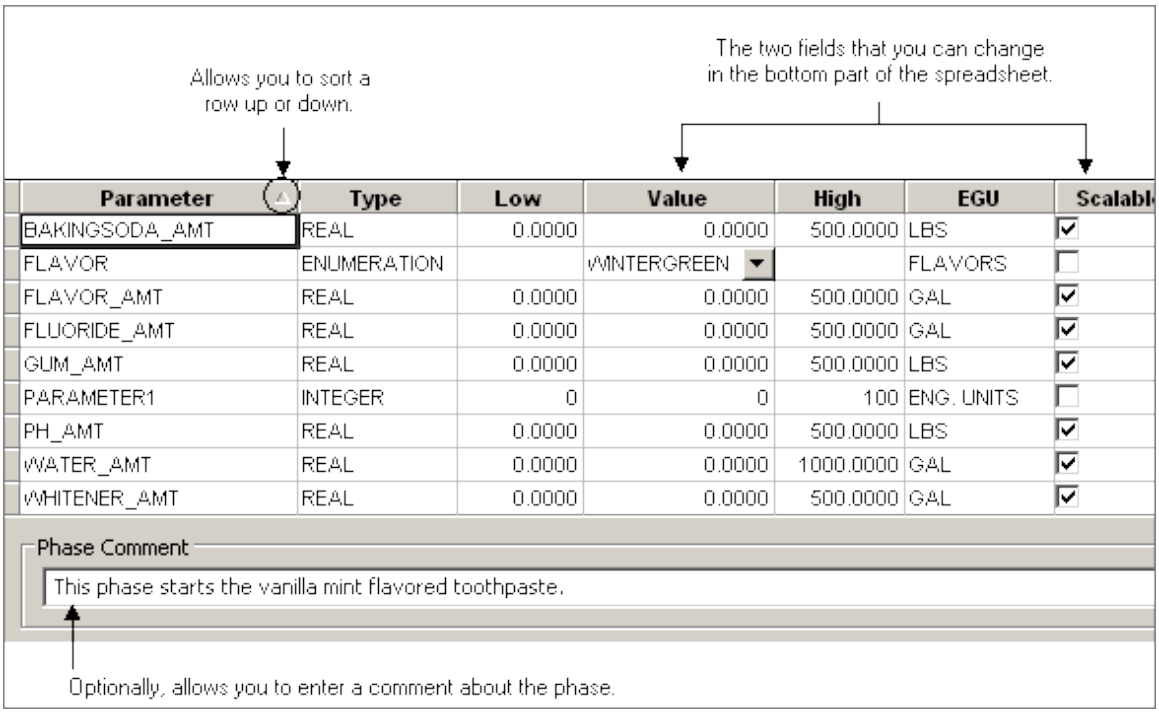


The following table provides more information on each of the parts of the top part of the Tabular Recipe Editor window.

Column	Description
Index	The sequence number that you assign to the recipe step. This number is 0 until you assign it a number. To assign or change a sequence number, click the Renumber Steps button. Or, on the Edit menu, click Renumber Steps.
Unit	The unit used by the recipe step. Only units used in the current area model are available.
Phase	The equipment phases available for the selected unit.
A	Path A. If you do not have parallel operations, for phases within a unit, this is the only path that will be selected.
B	Select this check box to identify a parallel path B.
C	Select this check box to identify a parallel path C.
D	Select this check box to identify a parallel path D.
E	Select this check box to identify a parallel path E.
S	Select this check box to split a recipe. This check box should be selected before you identify parallel steps. This option should only be used for equipment phases within the same unit.

Column	Description
J	Select this check box to join a recipe. This check box should be selected in the recipe step that appears after your parallel steps. This option should only be used for equipment phases within the same unit.
Description	Lists the parameters for the recipe. You can edit the values of these parameters using the parameter list in the bottom half of the recipe spreadsheet.

The next graphic illustrates the bottom half of the recipe spreadsheet, and how a specific unit's parameters appears in this part of the editor.



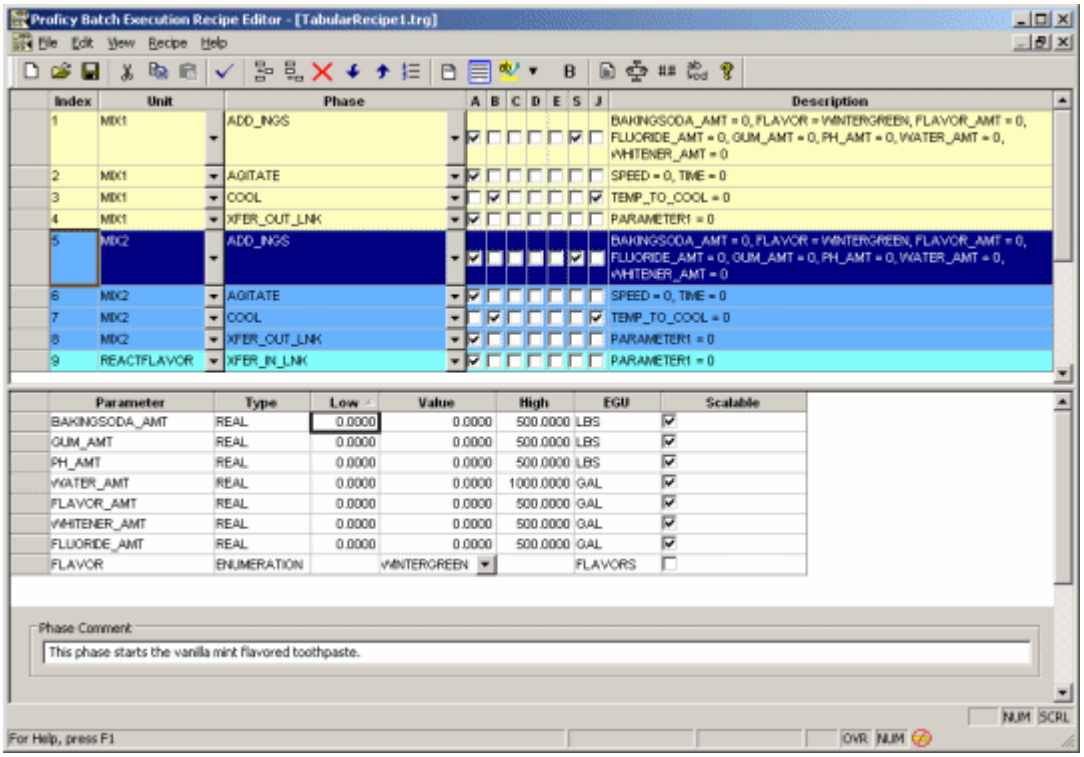
The following table provides more information on each of the parts of the bottom half of the Tabular Recipe Editor window.

Column	Description
Parameter	The name of the equipment phase parameter. This value cannot be edited in the Tabular Recipe Editor.
Type	The data type for the phase parameter: Real, Integer, String, Enumeration. This value cannot be edited in the Tabular Recipe Editor.
Low	The lowest value that this parameter can be set to. This value cannot be edited in the Tabular Recipe Editor.

Column	Description
Value	Indicates the current value for the phase parameter.
High	The highest value that this parameter can be set to. This value cannot be edited in the Tabular Recipe Editor.
EGU	The engineering units for the phase parameter. This value cannot be edited in the Tabular Recipe Editor.
Scalable	Indicates whether scaling at runtime is allowed.
Phase Comment	Allows for an optional comment for the phase parameter. The maximum number of characters that you can enter in this field is 1024.

Building a Recipe in the Tabular Recipe Editor

To build recipes in the Tabular Recipe Editor, you insert rows into a spreadsheet. To add rows, you can use the insert button on the toolbar, or the Insert or Append Recipe Step command in the Edit menu. Each row represents a step in the recipe. The following figure shows an example of how these rows appear in the Tabular Recipe Editor when you create your recipe:



Each row in the Tabular Recipe Editor contains a unit and associated phase. You can then define up to five parallel paths (A-E) for phases within a unit, and split and join these paths.

As with the SFC Recipe Editor, you can edit Value and Scalable parameters for each equipment phase. The Tabular Recipe Editor also allows you to use SEQUENCE parameter numbers, but you must configure these in the Parameter Name Settings dialog box.

A maximum of 2000 recipe steps (or rows) can be built into a single recipe in the Tabular Recipe Editor.

Formatting the Spreadsheet in the Tabular Recipe Editor

In the Tabular Recipe Editor, for each recipe you can change:

- The row color.
- Whether bold formatting is applied to the row text.

NOTE: *If you generate a report, this row color is used as the font color in the report. Similarly, if bold text is applied, bold text also appears in the report.*

These formatting features are available on the toolbar, or from the Edit menu. You can apply formatting to a single row or multiple rows. To select multiple rows, click and drag, and then apply the formatting.

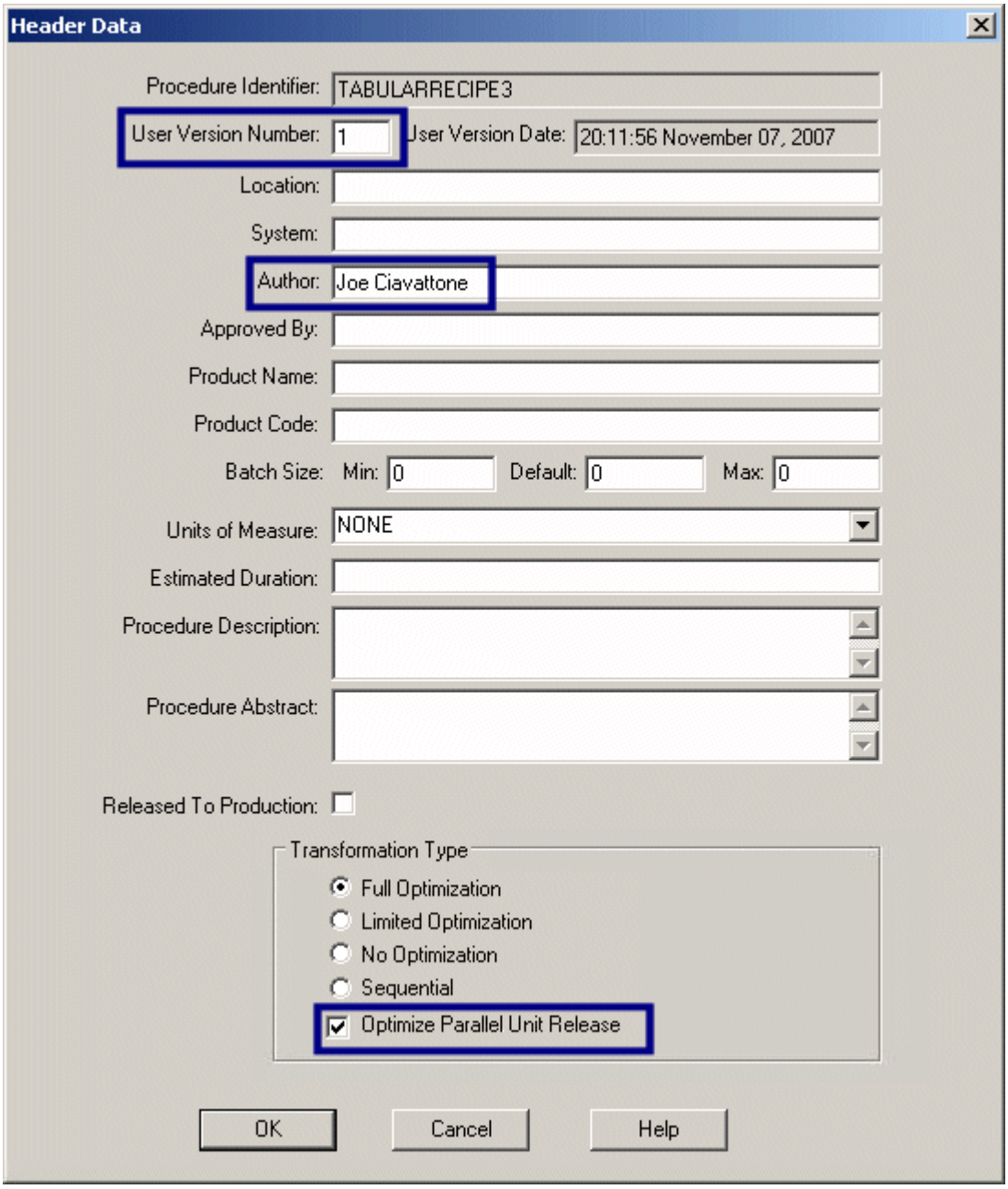
To pick another color, other than what's selected, on the Edit menu, click Pick Color. A standard Windows color picker is available. You can also view this color picker from the toolbar, by clicking the down arrow next to the Color button.

Recipe Header Data in the Tabular Recipe Editor

The recipe header contains administrative information for a recipe including the name, version number, author, version date, recipe description, and recipe batch size. Completing this information provides general information to anyone who wants to learn about and understand the recipe.

The Tabular Recipe Editor allows you to add header data to your recipe, just as the main, SFC Recipe Editor does. The Tabular Recipe Editor, however, includes a few extra fields such as Location and System, and the Optimize Parallel Unit Release option.

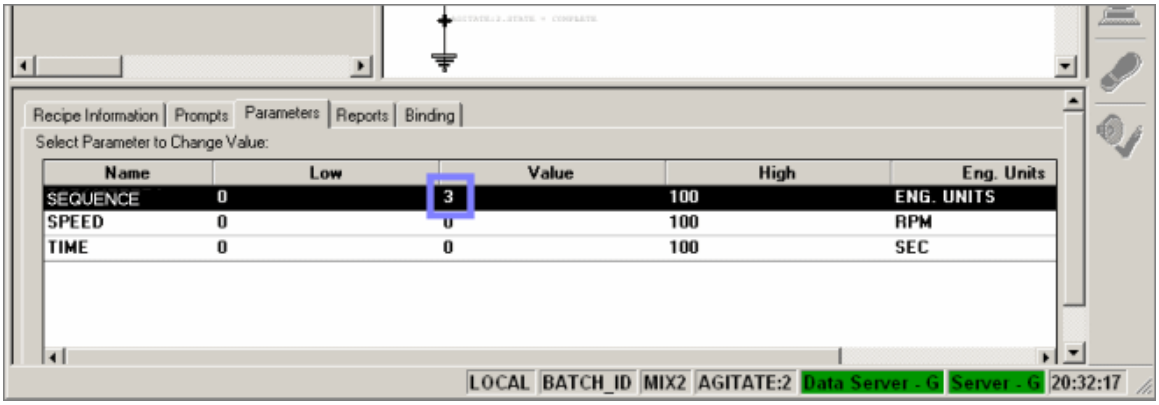
To access the Header Data dialog box, on the Recipe menu, click Recipe Header. It is recommended that you enter all required information, such as User Version Number and Author, along with the Min, Max, Default Size, and select whether or not you want to use the Optimize Parallel Unit Release option and the transformation type. All other fields in this dialog box are optional. Here is an example of the Recipe Header dialog box with these fields and check box highlighted:



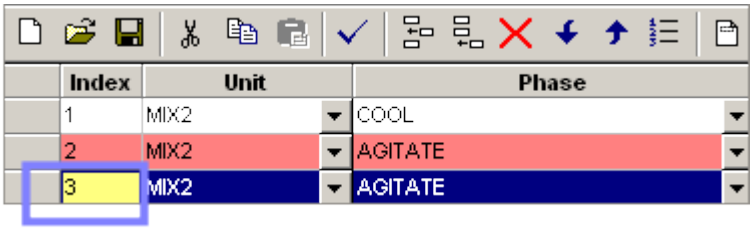
Both Recipe Editors require this information, along with the Version number, Version date, and Procedure identifier, to verify the recipe.

Sequence Number Parameters and the Tabular Recipe Editor

The Tabular Recipe Editor provides a way for you to track recipe step numbers in the Batch Execution Client when you are running a recipe. After you configure this feature, in the Procedure as SFC View in the Batch Client, you can select a phase and use the Parameters tab to view the sequence number. The following example shows a phase that is step number 3 in a tabular recipe:



This is step number 3 as it appears in the Tabular Recipe Editor:



As demonstrated, this feature allows you to easily track where a phase exists in a tabular recipe, or identify which tabular step is running in the SFC.

Configuring Sequence Numbers

To track sequence numbers, such as the SEQUENCE parameter in the above example, you need to do two things:

- In the Equipment Editor, add the sequence number parameter (for example, SEQUENCE), of type Integer that you configured in the Tabular Recipe Editor to each equipment phase class. The sequence parameter name has to be the same in all equipment phase classes. It is recommended that you add it to all classes, so that you can easily track any phase in the Batch Client.
- In the Tabular Recipe Editor, add a sequence number parameter to the tabular recipe. For example, SEQUENCE is the default sequence number parameter used in the example.

Searching and Replacing in the Tabular Recipe Editor

The Tabular Recipe Editor allows you to search for specific units, or phases within a unit, and then replace them with either a new unit or phase. On the Edit menu, choose the Search and Replace Units and Phases to open the Search and Replace dialog box.

With the Search and Replace dialog box, you can globally search and replace the entire spreadsheet, or just a few selected rows. To perform a global replacement, you select the Search and Replace Entire Recipe check box. To perform a search and replace on a few selected rows, click and drag to select the rows before opening the Search and Replace dialog box.

Revision History of a Recipe in the Tabular Recipe Editor

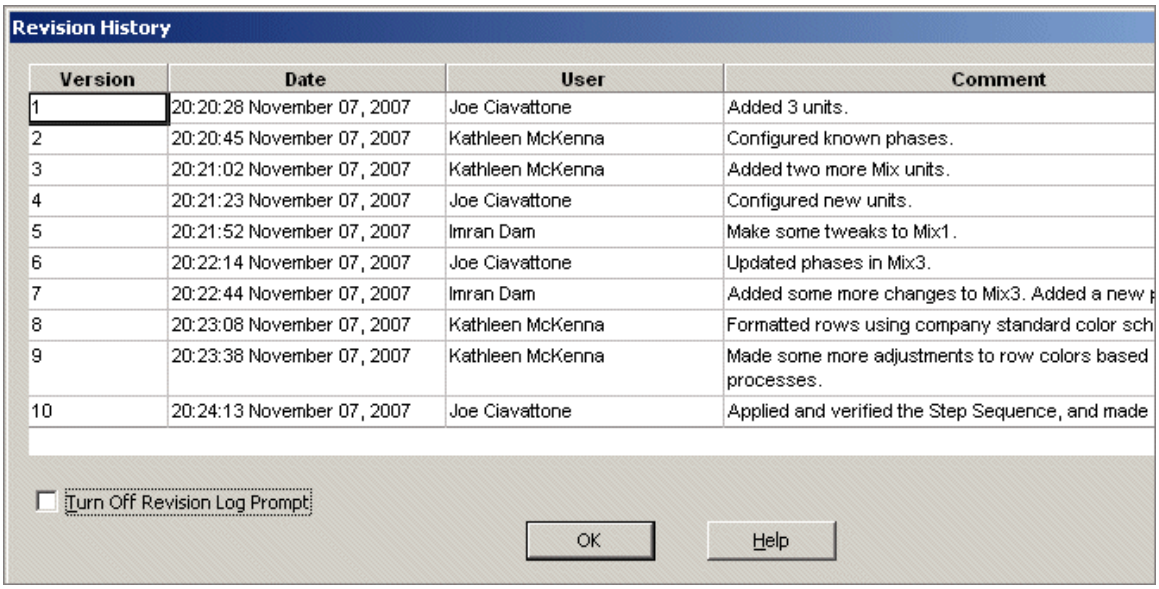
By default, the Enter Revision Log dialog box is displayed each time that you save a recipe or save it under a new name. The Enter Revision Log dialog box requests a:

- User name
- Revision number
- Comment

All fields are required, including the comment. You must manually update the version number, if required. The Tabular Recipe Editor does not increment the version number for you.

The Enter Revision Log dialog box can also be disabled. When disabled, a record is still written to the log. The record includes the current version number, the date, a user named and anonymous, and a blank comment. To re-enable this dialog box, you must open the Revision History window and clear the Turn Off Revision Log Prompt check box.

You can review each time that this recipe was saved, by reviewing the information in the recipe's Revision History window. To open this window in the Tabular Recipe Editor, on the Recipe menu, select View Revision History. The following figure shows an example of a revision log.



Importing and Exporting Recipes in the Tabular Recipe Editor

The Tabular Recipe Editor allows you to export or import recipes as .txt files, in tab delimited format. This allows for rapid development outside of the Tabular Recipe Editor, in another tool such as Microsoft Excel. It also provides a way in which you can DIFF recipe files, between versions, using your favorite DIFF tool.

The Export and Import commands are available on the File menu in the Tabular Recipe Editor.

A maximum of 2000 rows of data can be exported or imported into the Tabular Recipe Editor.

Report Generation in the Tabular Recipe Editor

The Tabular Recipe Editor allows you to generate a report of all steps in your tabular recipe. This report is saved to a file and viewed on screen. All color coding and bolded text used in your tabular recipe are represented in the report, as are the step numbers. You can change the page format of the report, zoom in, zoom out, or print the report, as shown in the following the figure.

To create a report, open your recipe and, on the File menu, select the Generate Report Option. By default, the report is saved to the ..\RECIPES\TabularRecipeReports folder. For example, for the DEMO project, this folder is: C:\Program Files\Proficy\Proficy Batch Execution\Projects\DEMO\RECIPES\TabularRecipeReports. A default name for the report includes the day and time that you ran the command.

Reports are as saved out as XML-formatted files, but appear with a .trr file extension.

Converting a Tabular Recipe to an SFC

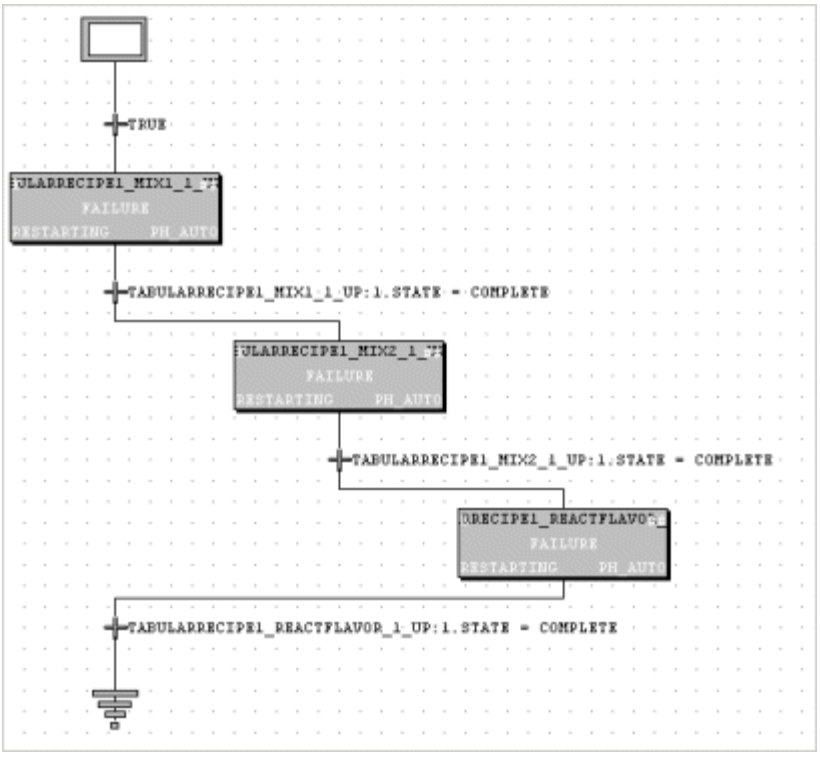
After you finish writing a recipe in the Tabular Recipe Editor, you should verify it, and then convert it to Sequential Function Chart (SFC) format. You need to generate this SFC format so that you can edit your recipe in the main Recipe Editor and then run your recipe in the Batch Client. Without the SFC format, you cannot run your tabular recipe in the Batch Client.

For example, if you have a recipe that looks like this in your Tabular Recipe Editor, with a split in step 1, and a join in step 3 (meaning steps 2 and 3 run in parallel). There is also a split at step 5 and a join at step 7 (meaning that steps 6 and 7 run in parallel). Splits and Joins should only be used for equipment

phases within the same unit. In the example below only MIX1 phases can appear between the Split and Join of steps 1 and 3.

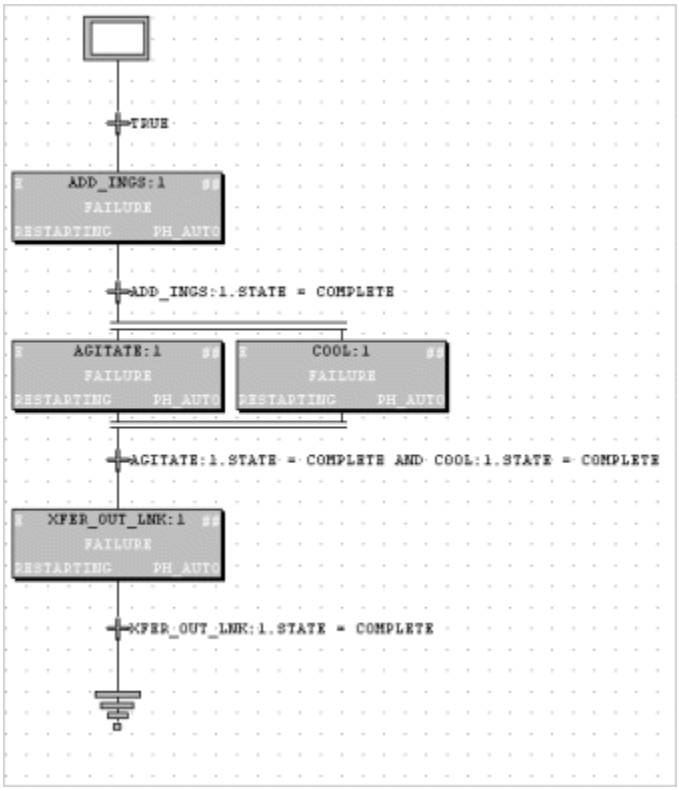
	Index	Unit	Phase	A	B	C	D	E	S	J
	1	MIX1	ADD_INGS	☑	☐	☐	☐	☐	☑	☐
	2	MIX1	AGITATE	☑	☐	☐	☐	☐	☐	☐
	3	MIX1	COOL	☐	☑	☐	☐	☐	☐	☑
	4	MIX1	XFER_OUT_LNK	☑	☐	☐	☐	☐	☐	☐
	5	MIX2	ADD_INGS	☑	☐	☐	☐	☐	☑	☐
	6	MIX2	AGITATE	☑	☐	☐	☐	☐	☐	☐
	7	MIX2	COOL	☐	☑	☐	☐	☐	☐	☑
	8	MIX2	XFER_OUT_LNK	☑	☐	☐	☐	☐	☐	☐
	9	REACTFLAVOR	XFER_IN_LNK	☑	☐	☐	☐	☐	☐	☐

After you select, on the File menu, the Transform Recipe to Batch SFC System command, an SFC is generated for the recipe. The *.UPC, *.UOP, *.BPC files are generated in your project's RECIPE's folder so that you can open the SFC in the main Recipe Editor. The steps above will appear similar to this when displayed as an SFC in your main Recipe Editor:



Batch Procedure Recipe Generated for Above Example

This is how parallel steps 2 and 3 appear at the phase level in the SFC view:



Operation Procedure Recipe Generated for Unit Named Mix1

Optimizing Parallel Unit Release for Tabular Recipes

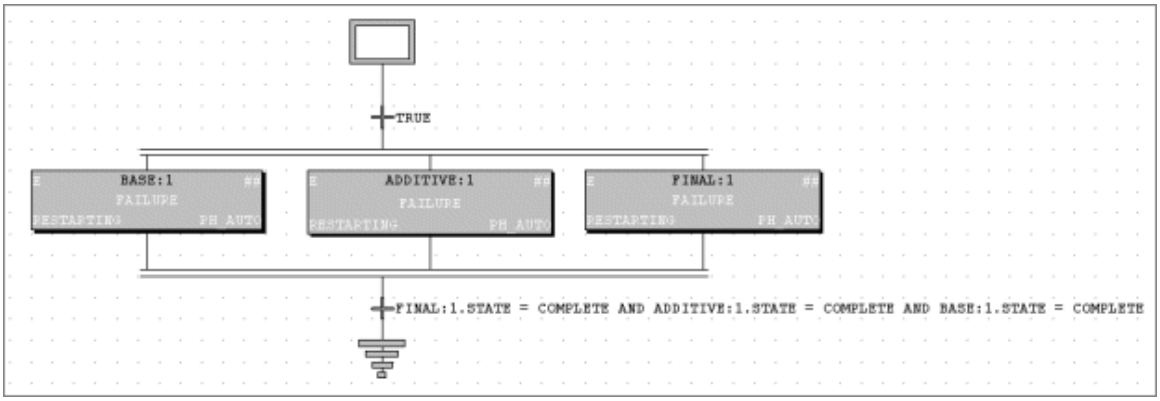
For tabular recipes, you can configure your recipe to release a unit when a unit procedure completes, rather than waiting for other parallel unit procedures to complete in the recipe. This allows units to be allocated more efficiently. To accomplish this, you need to:

- Configure a unit named "Dummy" in your area model
- Configure your recipe header to allow Optimize Parallel Unit Release

When you convert a tabular recipe to SFC format, the "Dummy" units are automatically added to your SFC, as shown in the second example below. You do not need to add them manually in either Recipe Editor.

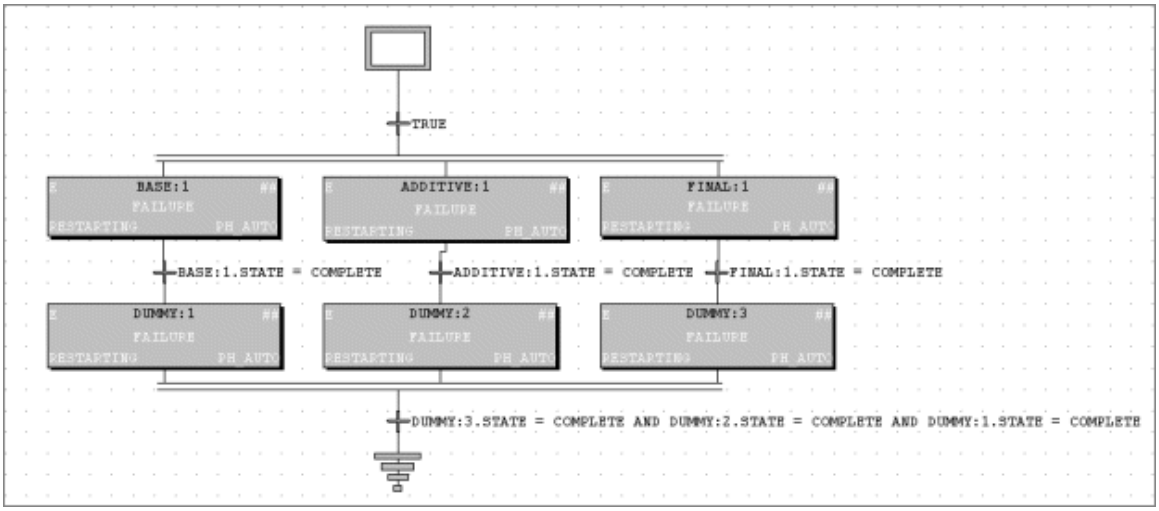
Optimize Parallel Unit Release Feature – Not Enabled

In the following recipe, all three units will not be released until the transition fires. The Optimize Parallel Unit Release feature is not enabled.



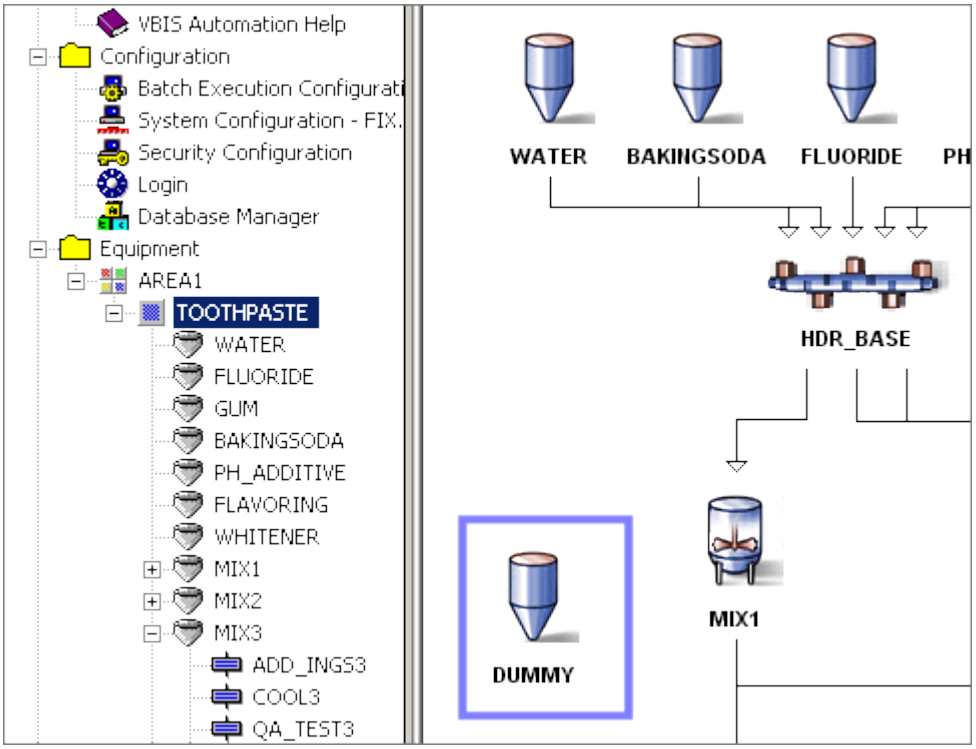
Optimize Parallel Unit Release Feature – Enabled

In this next example, there are three "Dummy" unit procedures, and the Optimize Parallel Unit Release *is* enabled. In this example, each unit is released as the last unit procedure within the unit completes. Equipment is available as soon as possible. There is no waiting on other units to finish before a unit can be used again.

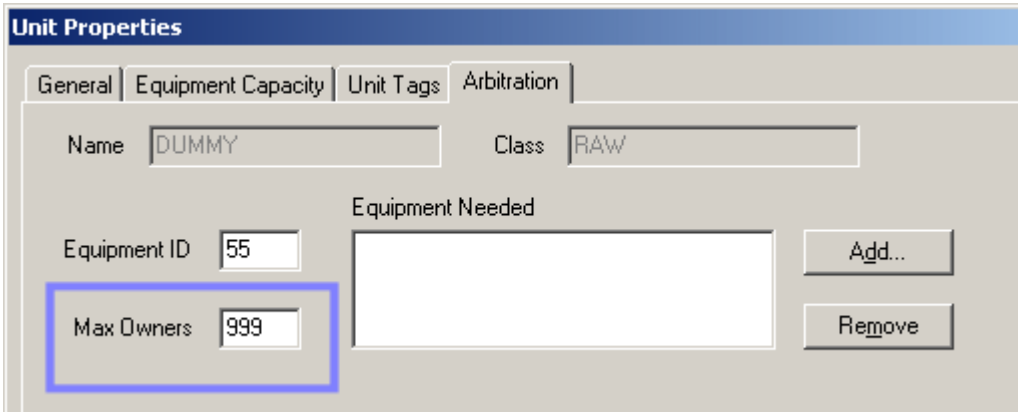


Configuration Overview

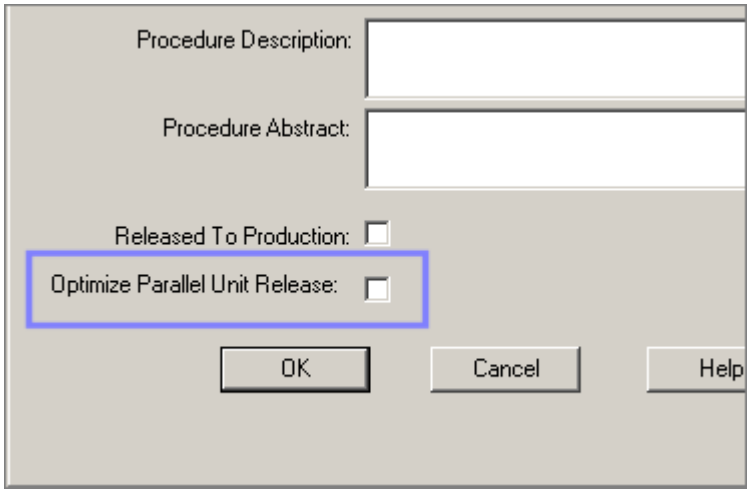
For the Optimize Parallel Unit Release feature to work properly, you must insert a unit named "Dummy" into your area model. The "Dummy" unit is not connected to other units, nor does it contain any phases, as shown in the following figure.



For the "Dummy" unit, you must set the Max Owners field in the Edit Equipment Phase dialog box to a larger number, such as 999, to allow for maximum simultaneous usage. Other fields can be left with the default settings, as shown in the following figure.



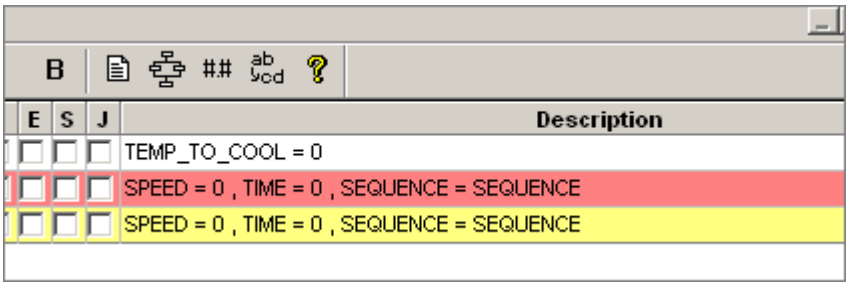
Finally, in the Tabular Recipe Editor you need to select the Optimize Parallel Unit Release check box in the Header Data dialog box, as shown in the following figure.



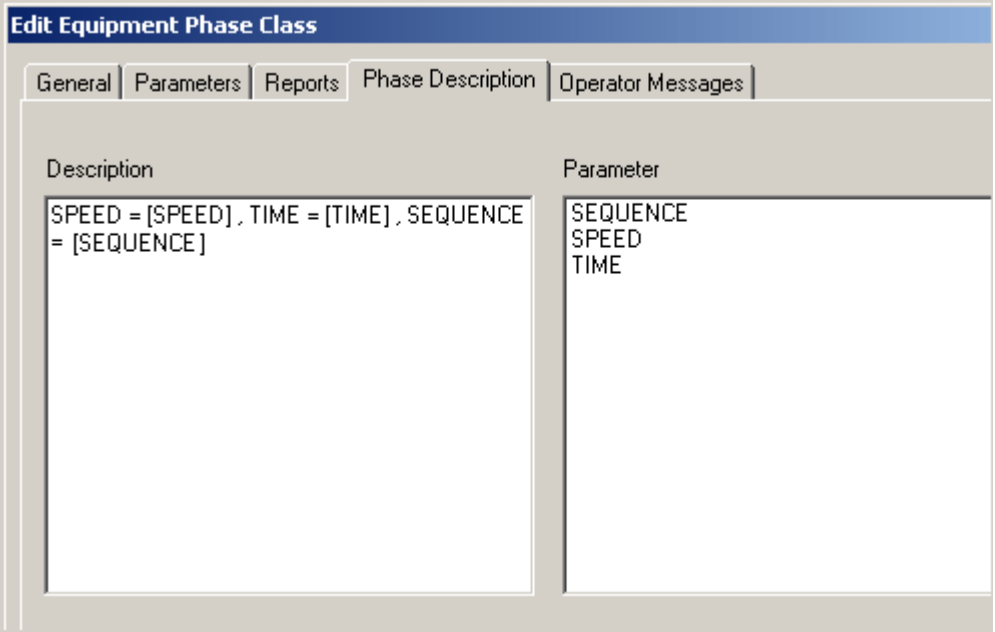
For more information on the Header Data dialog box, refer to the Recipe Header Data in the Tabular Recipe Editor section.

Enabling the Phase Description Tab in the Equipment Editor

Optionally, the Tabular Recipe Editor provides a way to change the text that appears in the Description field of the grid. The following example displays a recipe with three steps, and three Description fields:



In this example, for the step in the middle row, when you view the Phase Description tab on the Edit Equipment Phase Class dialog box in the Equipment Editor, you will notice that SPEED, TIME, and SEQUENCE are all listed in the Description area.



On the Phase Description tab, you can edit the text that appears in this Description area. The edits you make here will appear in the Tabular Recipe Editor's Description field the next time you open a recipe that uses this equipment phase class.

Main Recipe Editor Dialog Boxes

The main Recipe Editor application includes the following dialog boxes for working with recipes (listed in alphabetical order):

- Audit Information Dialog Box
- Edit Formulation Parameters for Recipe Dialog Box
- Electronic Work Instructions Dialog Box
- Equipment Requirements Dialog Box
- Export Recipe Dialog Box
- Font Dialog Box
- Header Data Dialog Box (Main Recipe Editor)
- Header Data Dialog Box (Product Formulations and Global Parameters)
- Key Process Reports Dialog Box
- New Dialog Box
- Operation Select Dialog Box
- Override Equipment Requirements Dialog Box
- Phase Link Group Dialog Box

- Print Dialog Box
- Print Setup Dialog Box
- Recipe Open Dialog Box
- Recipe Parameters Dialog Box
- Recipe Save As Dialog Box
- Recipe Storage Conversion Dialog Box
- Remove Recipe Dialog Box
- Select Phase Dialog Box
- Step Parameters Dialog Box
- Transition Expression Builder Dialog Box
- Unit Procedure Binding Dialog Box
- Unit Procedure Select Dialog Box
- Verification Results Dialog Box
- Zoom Dialog Box

Audit Information Dialog Box

The Audit Information dialog box displays the following items:

Audit Version

Displays the audit version number of the currently selected recipe in the Recipe tree. The version number increases by one each time the recipe is saved.

Unique Identifier (GUID)

Displays the unique, system-generated identifier for the recipe. Proficy Batch Execution generates the GUID when you create a new recipe for the first time, or use the Save As command to create a new recipe.

Performed By

The following table lists the contents of the Performed By area:

Item	Description
User Name	Displays the Microsoft Windows user ID of the operator (from the Performed By group) who last authorized the saving of the recipe. Information only appears in this field if you enabled auditing and configured Save or Save As signature requirements in the Proficy Batch Execution WorkSpace, and then saved the recipe with the required electronic signatures in the Proficy Batch Execution Recipe Editor.
Full Name	Displays the full user name of the operator (from the Performed By group) who last authorized the saving of the recipe. Information only appears in this field if you enabled auditing and configured Save or Save As signature requirements in the Proficy Batch Execution WorkSpace, and then saved the recipe with the required electronic signatures in the Proficy Batch Execution Recipe Editor.
Timestamp	Displays the date and time when Proficy Batch Execution authenticated the electronic signature of the operator (from the Performed By group). Information only appears in this field if you enabled auditing and configured Save or Save As signature requirements in the Proficy Batch Execution WorkSpace, and then saved the recipe with the required electronic signatures in the Proficy Batch Execution Recipe Editor.
Comment	Displays any comments entered by the operator (from the Performed By group) who last authorized the saving of the recipe. Information only appears in this field if you enabled auditing and configured Save or Save As signature requirements in the Proficy Batch Execution WorkSpace, and then saved the recipe with the required electronic signatures in the Proficy Batch Execution Recipe Editor.

Verified By

The following table lists the contents of the Verified By area:

Item	Description
User Name	Displays the Windows user ID of the supervisor (from the Verified By group) who last authorized the saving of the recipe. Information only appears in this field if you enabled auditing and configured Save or Save As signature requirements in the Proficy Batch Execution WorkSpace, and then saved the recipe with the required electronic signatures in the Proficy Batch Execution Recipe Editor.
Full Name	Displays the full user name of the supervisor (from the Verified By group) who last authorized the saving of the recipe. Information only appears in this field if you enabled auditing and configured Save or Save As signature requirements in the Proficy Batch Execution WorkSpace, and then saved the recipe with the required electronic signatures in the Proficy Batch Execution Recipe Editor.
Timestamp	Displays the date and time when Proficy Batch Execution authenticated the electronic signature of the supervisor (from the Verified By group). Information only appears in this field if you enabled auditing and configured Save or Save As signature requirements in the Proficy Batch Execution WorkSpace, and then saved the recipe with the required electronic signatures in the Proficy Batch Execution Recipe Editor.
Comment	Displays any comments entered by the supervisor (from the Verified By group) who last authorized the saving of the recipe. Information only appears in this field if you enabled auditing and configured Save or Save As signature requirements in the Proficy Batch Execution WorkSpace, and then saved the recipe with the required electronic signatures in the Proficy Batch Execution Recipe Editor.

Area Model Audit Information

The following table lists the contents of the Area Model Audit Information area:

Item	Description
File Name	Displays the full path name of the currently configured area model file. Example: C:\Program Files\Proficy\Proficy Batch Execution\Projects\DEMO\RECIPES\DEMO.CFG

Item	Description
Audit Version	Displays the audit version number of the currently configured area model file. The version number increases by one each time the area model is saved.
Unique Identifier (GUID)	Displays the system-generated, unique identifier for the area model. Proficy Batch Execution generates the GUID when you save a new area model for the first time, or use the Save As command to resave a area model.

Edit Formulation Parameters for Recipe Dialog Box

The Edit Formulation Parameters for Recipe dialog box displays the following items:

Tree Area

Displays the recipe procedure, unit procedure, or operation and its associated formulations and global parameter sets in tree view.

Date

Displays the date the selected formulation or global parameter set last changed. For example, a valid date might be: 11:54:31 Monday, June 23, 2008.

Version

Displays the internal version number of the selected formulation or global parameter set.

Author

Displays the name of the person who created this selected formulation or global parameter set.

Product Code

Displays the product code for the selected formulation.

Status

Displays the status of the selected formulation.

Description

Displays the description of the selected formulation.

Electronic Work Instructions Dialog Box

The Electronic Work Instructions dialog box displays the following items:

EIB List

Displays a list of EIBs associated with a selected phase. For each EIB, you can enter the name and version number.

New

Click to add an EIB file name.

Delete

Select an EIB in the list and click this button to remove it from the list.

Equipment Requirements Dialog Box

The Equipment Requirements dialog box displays the following items:

Equipment

The following table lists the contents of the Equipment area:

Item	Description
Class Based	Click this button if the unit procedure is a class-based recipe.
Unit Class	Select a unit class on which this class-based recipe will run.
Default Unit	Select the default unit to bind with this class-based unit procedure during batch execution.
Instance Based	Click this button if the unit procedure is an instance-based recipe.
Unit	Select the unit on which this instance-based recipe will run. Unit list box; Equipment Requirements dialog box New v4.0

Unit Procedure Capacity

The following table lists the contents of the Unit Procedure Capacity area:

Item	Description
Amount	Enter the minimum equipment capacity requirement for the unit procedure. For example, if the amount is 500 Liters, then the smallest possible unit this unit procedure can run on is a 500 Liter unit. Valid Entries: 0 - 9 and decimal point; up to 7 digits
UOM	Select a unit of measure for the required unit capacity value in the amount field. ***IMPORTANT:*** *The Unit of Measure defined here should match the UOM defined for the unit in the area model. If the UOMs do not match, capacity is not evaluated during Active Binding.*
Scale Capacity Requirement	Check to enable scaling of the capacity requirement. If scaling is enabled and the batch is scaled, the capacity is scaled by the same percentage as the batch.

Unit Binding

The following table lists the contents of the Unit Binding area:

Item	Description
Automatic Binding	Allows the Batch Execution Server to dynamically bind a unit to the unit procedure at runtime.
Operator Prompt	Prompts the operator to bind a unit procedure to a unit during runtime. ***NOTE:*** *The operator is prompted just before the unit procedure executes.*
Specify at Batch Creation	Requires the operator to specify unit binding at batch creation. (The operator is prompted as the batch is added to the Batch List.)
Modify Binding At Batch Creation	Allows the operator to bind units to unit procedures at batch creation.

Item	Description
Modify Binding During Batch Execution	Allows the operator to modify binding during batch execution. ***NOTE:*** *The operator can modify the binding of a unit to a unit procedure up until the transition preceding the unit procedure executes.*

Export Recipe Dialog Box

The Export Recipe dialog box displays the following items:

Save In

Specifies the location where you want to locate a file or folder. Click the arrow to select another location, or click the icon on the toolbar to move up levels. To open a file, in the Open dialog box, double-click the name of the file you want to open. To save a file using an existing file name, in the Save dialog box, double-click the name of the file you want to save.

Shortcuts

Provides shortcuts to places on your computer or the network from which you can open a file, such as the History folder, the desktop, or My Network Places. When you click a location, it will appear in Look in, and the files and folders in the selected location will be listed at the right.

Location

Lists the folders and files in the selected location.

File Name

Provides a space for you to type the name of the file you want to open or save. To quickly find a file you've previously opened, click the file name in the drop-down list, if available.

If you are searching for a file, you can use asterisks (*) as wildcards. For example, you can type *.* to see a list of all files. You can also type the full path of a file, for example, C:\Mydocs\Letter.doc.

If you are saving a file, you cannot use a question mark (?) or an asterisk in the file name. If you use a question mark or asterisk and click Save, the file will not be saved and the dialog box will not close.

Save As Type

Specifies the type of file you are saving.

Font Dialog Box

The Font dialog box displays the following items:

Font

Lists the available fonts.

Font Style

Lists the available styles for the specified font.

Size

Lists the available point sizes for the specified font.

Effects

Specifies whether to strike out or underline the font, and specifies the available colors for the font.

Sample

Shows a sample of how text will appear with the specified font settings.

Script

Lists the available language scripts for the specified font. When you select a different language script, the character set for that language becomes available for creating multilingual documents.

Header Data Dialog Box (Main Recipe Editor)

The Header Data dialog box displays the following items:

Procedure Identifier

Displays the name of the recipe.

Valid Entries: Up to 40 characters.

Examples: MAKE_TOOTHPASTE, ADDITIVE, QA_TEST

User Version Number

Enter the recipe version number. This field needs to be manually updated when modifying a recipe.

Valid Entries: Up to four characters or display only. The version number 1.0 appears in the field by default.

Examples: 2.35, 5.8a

User Version Date

Displays the time and date the recipe was last saved.

Valid Entries: Display only.

Example: 12:00:05 October 01, 2001

Author

Enter the name of the person who wrote the recipe.

Valid Entries: Up to 80 characters or display only.

Approved By

Enter the name of the person who approved the recipe.

Valid Entries: Up to 80 characters.

Product Name

Enter the name of the product manufactured with this recipe.

Valid Entries: Up to 80 characters.

Examples: Baking soda toothpaste, vanilla ice cream, chocolate chip cookies

Product Code

Enter the code assigned to the product.

Valid Entries: Up to 80 characters or display only.

Batch Size

The following table lists the contents of the Batch Size area:

Item	Description
Min	Enter the recipe's minimum batch size. Valid Entries: A numeric value greater than or equal to zero. You can enter a number between 0 and the maximum float data size of 3.402823466e+38F. Example: 1000

Item	Description
Default	Enter the recipe's default batch size. Valid Entries: Any numeric value between the Minimum and Maximum values, or equal to the Minimum or Maximum value. Example: 1000
Max	Enter the recipe's maximum batch size. Valid Entries: A numeric value greater than or equal to zero. You can enter a number between 0 and the maximum float data size of 3.402823466e+38F. Examples: 2000

Units of Measure

Enter the recipe's unit of measure.

Valid Entries: Any items appearing in the drop-down list.

Examples: Lbs., Gallons

Estimated Duration

Enter the time it takes to run a batch with this recipe.

Valid Entries: Up to 80 characters. Due to readability issues, however, it is recommended that you do not use the full 80 characters.

Examples: 25 minutes, 3 hours, 3 hrs and 30 min

Procedure Description

Enter text describing the recipe.

Valid Entries: Up to 132 characters or display only.

Procedure Abstract

Enter a summary of how the recipe works and what the recipe does.

Valid Entries: Up to 255 characters or display only.

Released to Production

Releases the recipe to production. Once released to production, operators can schedule and run a batch with this recipe.

File Name

Displays the file name of the current recipe.

Valid Entries: Display only.

Example: C:\Program Files\Proficy\Proficy Batch Execution\Projects\DEMO\RECIPES\MAKE_TOO.BPC

Storage Type

Displays how this recipe is stored, as a file on the local hard disk or in a SQL relational database.

Valid Entries: Display only.

Verification

The following table lists the contents of the Verification area:

Item	Description
Area Model File Name	Displays the full path name of the currently configured area model file. Example: C:\Program Files\Proficy\Proficy Batch Execution\Projects\DEMO\RECIPES\DEMO.CFG
Area Model File Timestamp	Displays the time and date that the area model was last saved. Example: 16:01:48 March 30, 2005
Time of Verification	Displays the time and date the recipe was last verified. Example: 12:30:49 April 02, 2005

Header Data Dialog Box (Product Formulations and Global Parameters)

The Header Data dialog box displays the following items:

Name

Enter a name for a formulation. This field is read-only for “Global_Parameters." This name must be unique to the project. You cannot have a duplicate formulation name within the entire project.

Version

Displays the internal version number of the formulation. Each time you save the formulation with a change, this value increments.

Version Date

Displays the date the formulation last changed. For example, a valid date might be: 11:54:31 Monday, June 23, 2008.

Author

Enter the name of the person who created this formulation.

Product Code

Optionally, enter the user-defined code assigned to this formulation. This field applies only to Product Formulations.

Description

Optionally, enter a description for this formulation. This field applies only to Product Formulations.

Status

Select a status for the formulation: Draft, Ready, Approved, or Withdrawn. This field applies only to Product Formulations. The values that appear in the drop-down list change depending upon the last value. For example, for a new formulation, you will only see Draft or Ready in this list. For a Ready formulation, Ready, Draft, and Approved appears in this list. For an Approved recipe, Approved, Ready, and Withdrawn appears in this list.

Key Process Reports Dialog Box

The Key Process Reports dialog box displays the following items:

Key

In the Key column, for each report that you want to make a key process report, select the corresponding check box.

Name

Displays the name of the report defined inside the equipment phase in the area model.

Type

Displays the data type of the report.

EGU

Displays the engineering units of the report.

New Dialog Box

The New dialog box displays the following item:

New

Displays the type of recipe you can create. By double-clicking one of these recipe types, you can create a new recipe.

Operation Select Dialog Box

The Operation Select dialog box displays the following items:

Recipe Name

Displays the available recipes. Double-click the recipe you want to select.

Author

Enter the name of the person who wrote the recipe.

Valid Entries: Up to 80 characters or display only.

Product Code

Enter the code assigned to the product.

Valid Entries: Up to 80 characters or display only.

Descr

Enter text describing the recipe.

Valid Entries: Up to 132 characters or display only.

Override Equipment Requirements Dialog Box

The Override Equipment Requirements dialog box displays the following items:

Equipment-Default Unit

Select the default unit to bind with this class-based unit procedure during batch execution.

Unit Procedure Capacity Group

The following table lists the contents of the Unit Procedure Capacity group:

Item	Description
Amount	Enter the minimum equipment capacity requirement for the unit procedure. For example, if the amount is 500 Liters, then the smallest possible unit this unit procedure can run on is a 500 Liter unit. Valid Entries: 0 - 9 and decimal point; up to 7 digits
UOM	Select a unit of measure for the required unit capacity value in the amount field. ***IMPORTANT:*** *The Unit of Measure defined here should match the UOM defined for the unit in the area model. If the UOMs do not match, capacity is not evaluated during Active Binding.*
Scale Capacity Requirement	Shows the status of the batch server.

Unit Binding Group

The following table lists the contents of the Unit Binding group:

Item	Description
Automatic Binding	Allows the Batch Execution Server to dynamically bind a unit to the unit procedure at runtime.
Operator Prompt	Prompts the operator to bind a unit procedure to a unit during runtime. ***NOTE:*** *The operator is prompted just before the unit procedure executes.*

Item	Description
Specify at Batch Creation	Requires the operator to specify unit binding at batch creation. (The operator is prompted as the batch is added to the Batch List.)
Modify Binding At Batch Creation	Allows the operator to bind units to unit procedures at batch creation.
Modify Binding During Batch Execution	Allows the operator to modify binding during batch execution. ***NOTE:*** *The operator can modify the binding of a unit to a unit procedure up until the transition preceding the unit procedure executes.*

Override Unit Procedure Equipment Requirements

Check to override the equipment requirements for the selected step.

Fetch Values

Click to restore the equipment requirements originally configured for this unit procedure.

Phase Link Group Dialog Box

The Phase Link Group dialog box displays the following items:

List Box

Displays the list of phase links.

Previous Group

Click this button to display the previous phase link group.

Add

Click this button to add the selected phase to the current phase link group.

Delete

Click this button to delete the selected phase from the selected link group.

Next Group

Click this button to display the next phase link group.

Print Dialog Box

The Print dialog box displays the following items:

Printer

The following table lists the contents of the Print area:

Item	Description
Name	Lists the printers that are connected to your computer.
Properties	Click to set up options for the printer. The options available depend on the printer's features.
Printer Information	Shows information about the selected printer, such as the printer's name and location.
Print to file	Prints the document to a file instead of routing it directly to a printer. The document is saved with the printer formatting, such as font selection and color specifications, in a .prn file that can be printed on another printer.

Print Range

Specifies whether to print the entire document, specific pages, or the portion you selected.

Copies Group

The following table lists the contents of the Copies group:

Item	Description
Number of Copies	Provides a space for you to type the number of copies you want to print.
Collate	If you have selected more than one copy, specifies whether you want the copies to be collated.

Print Setup Dialog Box

The Print Setup dialog box displays the following items:

Name

Lists the printers that are connected to your computer.

Properties

Click to set up options for the printer. The options available depend on the printer's features.

Paper

The following table lists the contents of the Paper group:

Item	Description
Size	Specifies the size of the paper, envelope, or other print media you want to use.
Source	Specifies where the paper you want to use is located in the printer. Different printer models support different paper sources, such as the upper tray, envelope feed, and manual feed.

Orientation

The following table lists the contents of the Orientation group:

Item	Description
Portrait	Specifies whether the document should be printed with its top along the short side of the paper (Portrait).
Landscape	Specifies whether the document should be printed along the long side of the paper (Landscape).

Recipe Open Dialog Box

The Recipe Open dialog box displays the following items:

Recipe Name

Displays the available recipes. Double-click the recipe you want to select.

Equipment

The following table lists the contents of the Equipment area:

Item	Description
All	Click this button to display the recipes associated with all the available units and unit classes.
Unit Class	Click this button if the unit procedure is a class-based recipe.
Unit	Click this button if the unit procedure is an instance-based recipe.
List Filter	Filters the list of available recipes. By selecting a unit or unit class from the list, the Recipe Editor displays only the recipes associated with the selected unit or unit class. Example: Select a unit, say MIX1, to display only the recipes associated with that unit.

Audit Version

Displays the audit version number of the currently selected recipe in the Recipe tree. The version number increases by one each time the recipe is saved.

User Version Number

Enter the recipe version number. This field needs to be manually updated when modifying a recipe.

Valid Entries: Up to four characters or display only. The version number 1.0 appears in the field by default.

Examples: 2.35, 5.8a

List Recipes of Type

Filters the list of available recipes.

Example: By selecting Operation, the Recipe Editor displays only the available operations in the Recipe Name list box.

Version Date

Displays the time and date the recipe was last saved.

Valid Entries: Display only.

Example: 12:00:05 October 01, 2001

Author

Enter the name of the person who wrote the recipe.

Valid Entries: Up to 80 characters or display only.

Product Code

Enter the code assigned to the product.

Valid Entries: Up to 80 characters or display only.

Procedure Description

Enter text describing the recipe.

Valid Entries: Up to 132 characters or display only.

Class/Instance

Displays the unit or unit class associated with the selected recipe.

Valid Entries: Display only.

Recipe Type

Displays the type of recipe that is selected.

Valid Entries: Display only.

Examples: PROCEDURE, UNIT_PROCEDURE, OPERATION

Released To Production

Displays whether this recipe is released to production. Once released to production, operators can schedule and run a batch with this recipe.

Area Model File Name

Displays the path and file name of the area model used by this recipe.

Valid Entries: Display only.

Example: C:\Program Files\Proficy\Proficy Batch Execution\Projects\DEMO\RECIPES\DEMO.CFG

Area Model Verified Against

Displays the time and date that the area model was last saved.

Example: 16:01:48 March 30, 2005

Verification Time

Displays the time and date the recipe was last verified.

Example: 12:30:49 April 02, 2005

File Name

Displays the file name of the current recipe.

Valid Entries: Display only.

Example: C:\Program Files\Proficy\Proficy Batch Execution\Projects\DEMO\RECIPES\MAKE_TOO.BPC

Storage Type

Displays how this recipe is stored, as a file on the local hard disk or in a SQL relational database.

Valid Entries: Display only.

Recipe Parameters Dialog Box

The Recipe Parameters dialog box displays the following items:

List Box

Displays a list of recipe parameters

New

Click this button to add a new recipe parameter to the current recipe. The maximum number of recipe parameters that you can add is 500.

Recipe Save As Dialog Box

The Recipe Save As dialog box displays the following items:

Procedure Identifier

Enter a new name for the recipe.

Audit Version

Displays the audit version number. By default, this is set to 1.

User Version Number

Enter the recipe version number. This field needs to be manually updated when modifying a recipe.

Valid Entries: Up to four characters or display only. The version number 1.0 appears in the field by default.

Examples: 2.35, 5.8a

Storage Type

Displays the currently selected format for storing recipes. When the text File appears in the field, the Recipe Editor stores recipes in separate files on a hard disk. When the text SQL appears in the field, the Recipe Editor stores recipes to a SQL relational database.

Valid Entries: File or SQL.

User Version Date

Displays the time and date the recipe was last saved.

Valid Entries: Display only.

Example: 12:00:05 October 01, 2001

Author

Enter the name of the person who wrote the recipe.

Valid Entries: Up to 80 characters or display only.

Product Identification Code

Enter the code assigned to the product.

Valid Entries: Up to 80 characters or display only.

Procedure Description

Enter text describing the recipe.

Valid Entries: Up to 132 characters or display only.

Procedure Abstract

Enter a summary of how the recipe works and what the recipe does.

Valid Entries: Up to 255 characters or display only.

Recipe Storage Conversion Dialog Box

The Recipe Storage Conversion dialog box displays the following items:

Convert SQL based project to FILE based project

Select this check box to convert all the recipes in the current project from a relational database and store them in files on your hard disk.

Convert FILE based project to SQL based project

Select this check box to convert all the recipes in the current project from file-based recipes and store them in a relational database.

Remove Recipe Dialog Box

The Remove Recipe dialog box displays the following items:

Recipe Name

Displays the available recipes. Double-click the recipe you want to select.

Delete

Closes the dialog box and saves any changes you have made.

List Recipes of Type

Filters the list of available recipes.

Example: By selecting Operation, the Recipe Editor displays only the available operations in the Recipe Name list box.

Equipment

The following table lists the contents of the Equipment area:

Item	Description
All	Click this button to display the recipes associated with all the available units and unit classes.
Unit Class	Click this button if the unit procedure is a class-based recipe.
Unit	Click this button if the unit procedure is an instance-based recipe.
Recipe List	Filters the list of available recipes. By selecting a unit or unit class from the list, the Recipe Editor displays only the recipes associated with the selected unit or unit class. Example: Select a unit, say MIX1, to display only the recipes associated with that unit.

Audit Version

Displays the audit version number of the currently selected recipe in the Recipe tree. The version number increases by one each time the recipe is saved.

User Version Number

Enter the recipe version number. This field needs to be manually updated when modifying a recipe.

Valid Entries: Up to four characters or display only. The version number 1.0 appears in the field by default.

Examples: 2.35, 5.8a

User Version Date

Displays the time and date the recipe was last saved.

Valid Entries: Display only.

Example: 12:00:05 October 01, 2001

Author

Enter the name of the person who wrote the recipe.

Valid Entries: Up to 80 characters or display only.

Product Code

Enter the code assigned to the product.

Valid Entries: Up to 80 characters or display only.

Description

Enter text describing the recipe.

Valid Entries: Up to 132 characters or display only.

Class/Instance

Displays the unit or unit class associated with the selected recipe.

Valid Entries: Display only.

Recipe Type

Displays the type of recipe that is selected.

Valid Entries: Display only.

Examples: PROCEDURE, UNIT_PROCEDURE, OPERATION

Select Phase Dialog Box

The Select Phase dialog box displays the following items:

Phase List Box

Displays the available phases. Double-click a phase to select it.

Step Parameters Dialog Box

The Step Parameters dialog box displays the following items:

Step Parameters List

Displays the step parameters available.

Transition Expression Builder Dialog Box

The Transition Expression Builder dialog box displays the following items:

Current Transition Status

Displays the Boolean condition of the current transition. To change the condition, edit the text in the field.

Buttons Group

The following table lists the contents of the Buttons group:

Item	Description
+	Click this button to add a plus sign (+) to the Boolean condition.
-	Click this button to add a minus sign (-) to the Boolean condition.
*	Click this button to add a multiplication sign (*) to the Boolean condition.
/	Click this button to add a division sign (/) to the Boolean condition.
<	Click this button to add a less than sign (<) to the Boolean condition.
<=	Click this button to add a less than or equal sign (<=) to the Boolean condition.
>	Click this button to add a greater than sign (>) to the Boolean condition.
>=	Click this button to add a greater than or equal sign (>=) to the Boolean condition.
=	Click this button to add an equal sign (=) to the Boolean condition.
<>	Click this button to add a not equal to sign (<>) to the Boolean condition.

Item	Description
(	Click this button to add a left parenthesis [( ] to the Boolean condition. Expressions within parentheses are evaluated before expressions outside of parentheses.
)	Click this button to add a right parenthesis [ )] to the Boolean condition. Expressions within parentheses are evaluated before expressions outside of parentheses.
Not	Click this button to add a NOT operator to the Boolean condition.
And	Click this button to add an AND operator to the Boolean condition.
Or	Click this button to add an OR operator to the Boolean condition.
Paste	Click this button to paste text from the clipboard into the Boolean condition.
Undo	Use this button to undo the changes you have made to the transition.

Category List

Displays the categories of items or specifies parameters/attributes you can add to a Boolean condition. Double-click the parameter or attribute to add it to the Boolean condition.

Operator List

Displays the specific operators or the type of data selected. Double-click an item in the list box to add it to the Boolean condition.

Unit Procedure Binding Dialog Box

The Unit Procedure Binding dialog box displays the following items:

Unit Procedure Binding List

Displays the list of unit procedure bindings.

New

Click to add a new row.

Delete

Click to delete a row.

Unit Procedure Select Dialog Box

The Unit Procedure Select dialog box displays the following items:

Recipe Name

Displays the available recipes. Double-click the recipe you want to select.

New

Click this button to create a new unit procedure for the selected step.

Author

Enter the name of the person who wrote the recipe.

Valid Entries: Up to 80 characters or display only.

Product Code

Enter the code assigned to the product.

Valid Entries: Up to 80 characters or display only.

Descr

Enter text describing the recipe.

Valid Entries: Up to 132 characters or display only.

Acquire Unit Prior To Starting Unit Procedure

Acquires the associated unit prior to starting the selected unit procedure. In other words, the batch procedure will acquire its units before the unit procedure runs.

By default, this check box is selected. It is recommended that you leave this check box selected. When this check box is cleared, units are allocated as the unit procedure runs, not beforehand. This could be an issue if multiple batches are running.

Verification Results Dialog Box

The Verification Results dialog box displays the following items:

Verification Results List

Lists the results of the verification.

Recipe Being Verified

Displays the name of the current recipe being verified.

Number Of Recipes Processed

Displays the number of recipes processed during the verification.

Number Of Messages

Displays the number of messages that appear in the verification results list.

Save

Click to save the verification results to a text .log file.

Zoom Dialog Box

The Zoom dialog box displays the following items:

Scale Group

The following table lists the contents of the Scale group:

Item	Description
Scale Options: 50, 100, 150, 200, 250, & 300%	Select the percentage at which you want to display the sequential function chart. For example, click the 300% button to display the recipe at 300% of its actual size.
Custom	Enter the percentage at which you want to display the sequential function chart.

Default

Click this button to enter 100% into the field.

Tabular Recipe Editor Dialog Boxes

The Tabular Recipe Editor application includes the following dialog boxes for working with tabular recipes (listed in alphabetical order):

- Color Dialog Box
- Enter Revision Log Dialog Box
- Export to Tab Delimited Text File Dialog Box
- Header Data Dialog Box
- Import from Tab Delimited Text File Dialog Box
- Parameter Name Settings Dialog Box
- Save As Dialog Box
- Save Report Dialog Box
- Search and Replace Dialog Box
- Revision History Dialog Box

Color Dialog Box

The Color dialog box displays the following items:

Basic Colors

Displays the basic colors available. You can define a custom color by clicking the closest basic color and then defining a custom color by using the color matrix.

Custom Colors

Displays any custom colors you have already defined:

- To change a custom color, click it, and then click Define Custom Colors. When you have completed your changes, click Add to Custom Colors.
- To define a new custom color, click an empty custom color box, and then click Define Custom Colors. Define the new color, and then click Add to Custom Colors.

Define Custom Color >>

Click to display the color matrix where you can define custom colors. To define a custom color, click the color in the color matrix, and then click Add to Custom Colors. You can also define a custom color by entering values for hue, saturation, and luminosity, or red, green, and blue, and then clicking Add to Custom Colors.

Enter Revision Log Dialog Box

The Enter Revision Log dialog box displays the following items:

User

Displays the revision history of the recipe. Each time you use the Save or Save As command, the revision is logged in this window. If Anonymous is logged as the user name, this means you selected the Turn Off Revision History check box so that a dialog box requesting the user name did not appear when the change was made.

Version

Enter a version number for this recipe.

Comment

Enter a comment to include with this revision. The comment is saved in the revision log. Up to 256 characters can be entered in this field.

Turn Off Revision Log Prompt

Select this check box if you do not want the Tabular Recipe Editor to prompt you for a user name whenever you save a recipe. In this case, Anonymous is logged as the user name in the Revision History window when you save a file. By default, this check box is not selected. When you clear this check box, every time you use the Save or Save As command, the Enter Revision Log dialog box appears requesting that you enter a user name, version, and comment.

Export to Tab Delimited Text File Dialog Box

The Export to Tab Delimited Text File dialog box displays the following items:

Save In

Specifies the location where you want to locate a file or folder. Click the arrow to select another location. The box below lists the items in the selected location.

File name

Provides a space for you to type the name of the file you want to save.

Save as Type

Specifies the type of file (*.txt) you are saving.

Header Data Dialog Box (Tabular Recipe Editor)

The Header Data dialog box displays the following items:

Procedure Identifier

Displays the name of the recipe.

Valid Entries: Up to 40 characters.

Examples: MAKE_TOOTHPASTE, ADDITIVE, QA_TEST

User Version Number

Enter the recipe version number. This field needs to be manually updated when modifying a recipe.

Valid Entries: Up to four characters or display only. The version number 1.0 appears in the field by default.

Examples: 2.35, 5.8a

User Version Date

Displays the time and date the recipe was last saved.

Valid Entries: Display only.

Example: 12:00:05 October 01, 2001

Location

Optionally, enter the name of the location where this recipe is run. You can enter up to 60 characters. This field is useful for plants with multiple sites.

System

Optionally, enter the name of the system from which this recipe is run. You can enter up to 60 characters. This field is useful for plants with multiple sites.

Author

Enter the name of the person who wrote the recipe.

Valid Entries: Up to 80 characters or display only.

Approved By

Enter the name of the person who approved the recipe.

Valid Entries: Up to 80 characters.

Product Name

Enter the name of the product manufactured with this recipe.

Valid Entries: Up to 80 characters.

Examples: Baking soda toothpaste, vanilla ice cream, chocolate chip cookies

Product Code

Enter the code assigned to the product.

Valid Entries: Up to 80 characters or display only.

Batch Size

The following table lists the contents of the Batch Size area:

Item	Description
Min	Enter the recipe's minimum batch size. Valid Entries: A numeric value greater than or equal to zero. You can enter a number between 0 and the maximum float data size of 3.402823466e+38F. Example: 1000
Default	Enter the recipe's default batch size. Valid Entries: Any numeric value between the Minimum and Maximum values, or equal to the Minimum or Maximum value. Example: 1000
Max	Enter the recipe's maximum batch size. Valid Entries: A numeric value greater than or equal to zero. You can enter a number between 0 and the maximum float data size of 3.402823466e+38F. Examples: 2000

Units of Measure

Enter the recipe's unit of measure.

Valid Entries: Any items appearing in the drop-down list.

Examples: Lbs., Gallons

Estimated Duration

Enter the time it takes to run a batch with this recipe.

Valid Entries: Up to 80 characters. Due to readability issues, however, it is recommended that you do not use the full 80 characters.

Examples: 25 minutes, 3 hours, 3 hrs and 30 min

Procedure Description

Enter text describing the recipe.

Valid Entries: Up to 132 characters or display only.

Procedure Abstract

Enter a summary of how the recipe works and what the recipe does.

Valid Entries: Up to 255 characters or display only.

Released To Production

Releases the recipe to production. Once released to production, operators can schedule and run a batch with this recipe.

Transformation Type

The Transformation Type area includes the following items:

Item	Description
Full Optimization	When you generate an SFC recipe, it will contain the units in parallel, only when there are dependencies. This setting creates the most compact grouping of phases, operations, and unit procedures. Full Optimization is the default setting.
Limited Optimization	When you generate an SFC recipe, it will contain all units in parallel, phases grouped in operations and unit procedures whenever possible.
No Optimization	When you generate an SFC recipe, it will contain all units in parallel. The phases are grouped only if they are contained within a split and join.

Item	Description
Sequential	When you generate an SFC recipe, it will contain no units in parallel. It will have sequential execution through units.
Optimize Parallel Unit Release	Select this check box if you want to release a unit when a unit procedure completes, without waiting for other parallel unit procedures to complete in the recipe. This allows units to be allocated more efficiently. For this check box to work, you must also insert a unit named "Dummy" in your area model. The "Dummy" unit is not connected to other units, nor does it contain any phases. It does, however, have the Max Owners field in the Edit Equipment Phase dialog box set to a larger number, such as 999, to allow for maximum simultaneous usage.

Import from Tab Delimited Text File Dialog Box

The Import from Tab Delimited Text File dialog box displays the following items:

Look in

Specifies the location where you want to locate a file or folder. Click the arrow to select another location. The box below lists the items in the selected location.

File name

Provides a space for you to type the name of the file you want to open.

Files of Type

Lists the types of files (*.txt) to display.

Open as Read-Only

Specifies that you can read the file but cannot make changes to it.

Parameter Name Settings Dialog Box

The Parameter Name Settings dialog box displays the following items:

Step Sequence Number

Leave the default of SEQUENCE, or enter another name for the sequence number parameter that you want to use with your recipe.

NOTE: *The parameter that you enter here must already be configured as a phase parameter (an Integer) for the equipment phase classes in the Batch Execution Equipment Editor, in the Edit Equipment Phase Class dialog box.*

Update Current Document Parameter Values

Click this button to apply the recipe parameter name that you entered in the Step Sequence Number field to your current recipe. Make sure that you validate this parameter name by clicking the Validate Parameter Names in Current Document button.

Validate Parameter Names In Current Document

Click this button to validate the recipe parameter you entered in the Step Sequence Number field.

Validation Results

This area displays any error messages that occur when you validate your sequence number parameter.

Save As Dialog Box

The Save As dialog box displays the following items:

Save In

Specifies the location where you want to locate a file or folder. Click the arrow to select another location. The box below lists the items in the selected location.

File name

Provides a space for you to type the name of the file you want to save.

Save as Type

Specifies the type of file (*.trg for tabular recipes) that you are saving.

Save Report Dialog Box

The Save Report dialog box displays the following items:

Save In

Specifies the location where you want to locate a file or folder. Click the arrow to select another location. The box below lists the items in the selected location.

File name

Provides a space for you to type the name of the file you want to save.

Save as Type

Specifies the type of file (*.trr for the reports) you are saving.

Search and Replace Dialog Box

The Search and Replace dialog box displays the following items:

Search And Replace Unit Group

The following table lists the contents of the Search And Replace Unit group:

Item	Description
Find Which Unit	Select a unit to search for in the drop-down list. ***NOTE:*** *If you want to search the entire recipe for a specific unit, select the Search and Replace Entire Recipe check box. Otherwise, only the selected rows are searched.*
Replace Unit With	Select a unit to search for in the drop-down list. ***NOTE:*** *If you want to search the entire recipe for a specific unit, select the Search and Replace Entire Recipe check box. Otherwise, only the selected rows are searched.*
Replace All	Click this button to make the replacements.

Search And Replace Phase Group

The following table lists the contents of the Search And Replace Phase group:

Item	Description
Find Which Unit	Select a unit to search for in the drop-down list. ***NOTE:*** *If you want to search the entire recipe for a specific unit, select the Search and Replace Entire Recipe check box.*
Find Which Phase	Select a phase to search for in the drop-down list. Only phases from the selected unit are available from this drop-down list.
Replace Phase With	Select the phase that you want to use as a replacement. Only phases from the selected unit are available from this drop-down list.
Replace All	Click this button to make the replacements.

Search and Replace Entire Recipe

Select this check box if you want to search the entire recipe (all units), instead of just the selected row or rows.

Warnings

This area displays all warning messages during the search and replace action.

Revision History Dialog Box

The Revision History dialog box displays the following items:

Revision History List

The following table lists the contents of the Revision History list:

Item	Description
Version	Displays the version number of the recipe. Every time you use Save or Save As for the selected recipe, the revision is recorded in this log. The revision information is saved as part of the tabular recipe file.
Date	Displays the date of the revision.
User	Displays the user name who performed the revision.

Item	Description
Comment	Display the comment (if any) entered along with the revision.

Turn Off Revision Log Prompt

Select this check box if you do not want the Tabular Recipe Editor to prompt you for a user name whenever you save a recipe. In this case, Anonymous is logged as the user name in the Revision History window when you save a file. By default, this check box is not selected.

When you clear this check box, every time you use the Save or Save As command, the Enter Revision Log dialog box appears requesting that you enter a user name, version, and comment.

How Do I...

The following sections explain how to work with the Recipe Editor:

- Building a Recipe
- Modifying a Recipe
- Deleting Recipes
- Working with Recipes

Building a Recipe

For information on building a recipe in the Recipe Editor, refer to the following sections:

- Overview: Building a Recipe
- Selecting the Recipe Level and the Equipment Requirements
- Using Forced Bindings
- Configuring a Recipe's Material Flow
- Constructing a Sequential Function Chart
- Recipe and Step Parameters
- Completing the Recipe Header
- Saving and Verifying Recipes
- Releasing a Recipe to Production
- Printing a Recipe
- Creating a Phase Link Group

Overview: Building a Recipe

To build a recipe:

1. In the Recipe Editor, on the toolbar, click the New button. The New dialog box appears.
2. Specify the type of recipe and its equipment requirements. This step lets you define the specific device (or unit) the recipe requires when it runs.
3. Build a sequential function chart for the selected procedure level.
4. Configure the material flow for the recipe.
5. Define any required recipe parameters.
6. Define any recipe ingredients.
7. Complete the recipe header. This step is required in order to verify the recipe.
8. Save and verify the recipe.
9. Release the recipe to production.

Selecting the Recipe Level and the Equipment Requirements

To select the level of a recipe and define the equipment requirements:

1. In the Recipe Editor, on the toolbar, click the New button. The New dialog box appears.
2. Enter any required electronic signatures.
3. Select the recipe level you want to create and click OK. If you selected unit procedure or operation as the recipe level, the Equipment Requirements dialog box appears.
4. Define the equipment requirements for the:
    - Unit procedure.
    - Operation.

Using Forced Bindings

To force unit procedures to run on the same or on different units:

1. In the Recipe Editor, on the toolbar, click the Forced Unit Procedures Binding button. The Unit Procedure Binding dialog box appears.
2. Click New to add a new row to the dialog box.
3. Select a unit procedure from the first column's list box.
4. Select a unit procedure from the second column's list box.
5. Select either the Same Unit or the Different Unit check box.
6. Click OK.

    NOTE: *The unit procedures must be configured to use the same unit class.*

Configuring a Recipe's Material Flow

To configure a recipe's material flow:

1. In the Recipe Editor, for your recipe, configure Jacobson Links.
2. Configure forced bindings.

Constructing a Sequential Function Chart

For information on constructing a sequential function chart in the Recipe Editor, refer to the following sections:

- Overview: Constructing Sequential Function Charts
- Adding Steps
- Configuring Steps
- Moving Steps
- Moving Multiple Steps
- Adding Transitions
- Configuring Transitions
- Linking Steps and Transitions
- Adding Jacobson Links
- Creating Loops
- Changing the Font

Overview: Constructing Sequential Function Charts

To construct sequential function charts:

1. In the Recipe Editor, create a recipe.
2. Add steps to each operation, unit procedure, and procedure.
3. Configure each step.
4. Add transitions.
5. Link each step and transition together.
6. Configure each transition.

Adding Steps

To add a step to a sequential function chart:

1. In the Recipe Editor, on the SFC toolbar, click the Step tool. The cursor changes to a right angle.
2. Position the cursor where you want the step to appear in the sequential function chart.

3. Click the left mouse button. A new step appears.
4. Repeat this procedure until you have added all the steps you need.

Configuring Steps

To configure a step in a sequential function chart:

1. In the Recipe Editor, on the SFC toolbar, click the Selection tool.
2. Double-click the step you want to configure. A list of the available phases, operations, or unit procedures appears, depending on the level of recipe you are creating.
3. Select a phase, operation, or unit procedure from the list.
4. Click OK. The text in the step changes to reflect your selection.

Moving Steps

To move a step in a sequential function chart:

1. In the Recipe Editor, on the SFC toolbar, click the Selection tool.
2. In the recipe, select the step you want to move.
3. Click and hold the left mouse button.
4. Drag the mouse to the location where you want the step to appear.
5. Release the mouse button.

TIP: *You can also move a step with the arrow keys. To do this, select the step you want to move. Press and hold an arrow key. The step moves in the direction indicated by the arrow key.*

Moving Multiple Steps

To move multiple steps in a sequential function chart:

1. In the Recipe Editor, on the SFC toolbar, click the Selection tool.
2. Click and hold the left mouse button in an unused portion of the screen.
3. Drag the mouse toward the steps you want to select. A selection box appears as you drag the mouse.
4. When the steps you want to select are inside the selection box, release the mouse button. The steps inside the selection box are selected.

   The selected items have blue highlights and dotted lines surrounding them.

5. Click and hold the left mouse button on one of the selected steps.
6. Drag the mouse to the location where you want the step to appear.
7. Release the mouse button.

   NOTE: *You must place the mouse pointer on one of the steps or transitions in the selection box in order to move all the selected items. If you don't, all items will be de-selected.*

To move multiple steps with the arrow keys:

1. Click and hold the left mouse button in an unused portion of the screen.
2. Drag the mouse toward the steps you want to select. A selection box appears as you drag the mouse.
3. When the steps you want to select are inside the selection box, release the mouse button. The steps inside the selection box are selected.
4. Press and hold an arrow key. The steps move in the direction indicated by the arrow key.

Adding Transitions

To add a transition to a sequential function chart:

1. In the Recipe Editor, on the SFC toolbar, click the Transition tool. The cursor changes to a plus sign.
2. Position the mouse pointer where you want to place the transition in the sequential function chart.
3. Click the left mouse button. The transition appears with its condition set to TRUE.
4. Make your step changes:

Configuring Transitions

To configure a transition:

1. In the Recipe Editor, in a recipe, double-click the transition you want to configure. The Transition Expression Builder dialog box appears.
2. Press Delete to clear the current condition and enter a new condition.
3. Use the row of buttons in the middle of the dialog box to add operators to a condition.
4. Use the list boxes at the bottom of the dialog box to select the items you want to add from a list. When using these list boxes:
    - You select an item from the left list box.
    - The categories of the selected item appear in the middle list box.
    - The items in the selected category appear in the right list box.
    - You can add an item to the condition at the top of the dialog box by double-clicking the item from the middle or right list box.
    - If you see a plus sign next to a folder in the left list box, there are items inside the folder. To open the folder, click the plus sign.

Linking Steps and Transitions

To link a step to a transition:

1. In the Recipe Editor, on the SFC toolbar, click the Link tool. The cursor changes to a link pointer.
2. Position the mouse pointer over the step you want to link.

3. Click and hold the left mouse button.
4. Drag the mouse from the step to the transition.
5. Position the mouse pointer over the transition and release the mouse button. A line appears connecting the step and transition.
6. Repeat steps 1 through 5 alternating between linking steps to transitions and transitions to steps until the entire sequential function chart is complete.

Adding Jacobson Links

To add Jacobson Links:

1. In the Recipe Editor, on the toolbar, click the Jacobson Link tool button.
2. Position the mouse pointer over the origin step.
3. Click and hold the left mouse button.
4. Drag the mouse from the origination step to the destination step.
5. Position the mouse pointer over the destination step and release the mouse button. A red, curved line (Jacobson Link) appears connecting the origin and the destination steps.

Creating Loops

To create a loop:

1. In the Recipe Editor, in a recipe, link your steps and transitions together until all the steps and transitions, except the looping transition, are connected.
2. Select the looping transition with the Link tool.
3. Click and hold the left mouse button.
4. Drag the mouse from the looping transition to the step it loops back to and release the mouse button. A line appears connecting the looping transition and the selected step.
5. Select the step preceding the concluding transition with the Link tool.
6. Click and hold the left mouse button.
7. Drag the mouse from the step to the looping transition and release the mouse button.

Changing the Font

To change the font of all steps in a sequential function chart:

1. In the Recipe Editor, on the Format menu, click Font. The Font dialog box appears.
2. Select the font you want to use from the Font list box.
3. Select the font style from the Font Style list box.
4. Select the font size from the Font Size list box.
5. Click OK to save your changes.

Recipe and Step Parameters

For information on recipe and step parameters in the Recipe Editor, refer to the following sections:

- Modifying Recipe Parameters
- Modifying Step Parameters
- Setting a Parameter's Value
- Specifying a Deferred Parameter
- Defining an Operator Parameter

Modifying a Recipe Parameter

To modify a recipe parameter for a procedure, unit procedure, or operation:

1. In the Recipe Editor, from within a recipe, right-click the recipe step, and select Recipe Parameters. The Recipe Parameters dialog box appears.
2. Optionally, in the Low field, enter the new low EGU value. This field is available only if the parameter is a Real or Integer type.
3. Optionally, in the High field, enter the new high EGU value. This field is available only if the parameter is a Real or Integer type..

   NOTE: *The low and high level ranges for the step parameters can only be reduced from the phase parameter ranges, not expanded.*
4. Optionally, in the Default field, enter the recipe parameter’s default value.
5. Optionally, select the Scaleable check box to make the parameter scaleable, as needed.

   NOTE: *You cannot scale enumerations or strings.*
6. Verify the recipe.

Modifying a Step Parameter

To modify step parameter information:

1. In the Recipe Editor, from within a recipe, right-click the step with the step parameter you want to edit, and click Step Parameters. The Step Parameters dialog box appears.
2. Select the row that you want to edit by placing your cursor in the Origin field.
3. In the Origin column of the selected row, make a selection from the drop-down list:
   - **Value** – Use a constant value for the step parameter.
   - **Defer** – Defer the value to the next level. This option is provided for backwards compatibility. When deferring parameters, it is recommended that you use the Defer to Schedule, Defer to Procedure, or Defer to Unit Procedure options, because you do not have to manually defer up each level (as you do when you select the Defer option with multiple levels).
   - **Defer to Schedule** – Defer the value to the schedule level. The value is provided when you schedule the recipe.

- **Defer to Procedure** – Defer the value to the procedure level.
- **Defer to Unit Procedure** – Defer the value to the unit procedure level.
- **Operator** – Allow the operator to provide the value.

4. In the Value field, enter a value for the parameter. For deferrals, the value you enter here is used to pre-populate the value field at the next level up.
5. Optionally, in the Low field, enter the new low EGU value. This field is available only if the parameter is a Real or Integer type.
6. Optionally, in the High field, enter the new high EGU value. This field is available only if the parameter is a Real or Integer type.

   NOTE: *The low and high level ranges for the step parameters can only be reduced from the phase parameter ranges, not expanded.*

7. If you want to display this parameter and its engineering units (EGU) in the SFC, select the Display check box. If step parameter is a deferred parameter, the SFC step displays the deferred parameter name.
8. Click OK.
9. Verify the recipe to update the step and recipe parameters, and then save it.

   IMPORTANT: *It is very important that you verify the recipe so that your step and recipe parameters are updated. Recipes will need to be verified to the highest level that you chose to defer. For example, if you chose to Defer to Procedure, you should verify the recipes at the Procedure level so the automatic propagation of the parameters will take place. If you select to Defer to Unit Procedure, you should verify the recipe at the Unit Procedure level.*

Setting a Constant Value for a Step Parameter

To define a constant value for a step parameter:

1. In the Recipe Editor, from within a recipe, right-click the step with the step parameter you want to edit, and click Step Parameters. The Step Parameters dialog box appears.
2. In the Value column of the spreadsheet, place your cursor in the Value field for the step parameter you want to modify.
3. In the Value field, enter a value for the parameter.

   NOTE: *The low and high level EGU ranges are not available for editing when Value appears in the Origin field.*

4. Click OK.
5. Verify the recipe to update the step and recipe parameters.

Defining a Deferred Parameter

To define a deferred parameter:

1. In the Recipe Editor, from within a recipe, right-click the step with the step parameter you want to edit, and click Step Parameters. The Step Parameters dialog box appears.
2. From the Origin column of the spreadsheet, select deferral option applicable to your recipe level:

- **Defer to Schedule** – Defer the value to the schedule level. The value is provided when you schedule the recipe.
- **Defer to Procedure** – Defer the value to the procedure level.
- **Defer to Unit Procedure** – Defer the value to the unit procedure level.

NOTE: *The "Defer" option is still available for deferring the value up to the next level. The Defer option is provided for backwards compatibility only. When deferring parameters, it is recommended that you use the Defer to Schedule, Defer to Procedure, or Defer to Unit Procedure options for convenience, because you do not have to manually defer up each level (as you do when you select the Defer option with multiple levels).*

3. In the Value field, enter a value for the parameter. The value you enter here is used to pre-populate the value field at the next level up.
4. Optionally, in the Low field, enter the new low EGU value. This field is available only if the parameter is a Real or Integer type.
5. Optionally, in the High field, enter the new high EGU value. This field is available only if the parameter is a Real or Integer type.

   NOTE: *The low and high level ranges for the step parameters can only be reduced from the phase parameter ranges, not expanded.*

6. Click OK.
7. Verify the recipe to update the step and recipe parameters, and then save it.

   IMPORTANT: *It is very important that you verify the recipe so that your step and recipe parameters are updated. Recipes will need to be verified to the highest level that you chose to defer. For example, if you chose to Defer to Procedure, you should verify the recipes at the Procedure level so the automatic propagation of the parameters will take place. If you select to Defer to Unit Procedure, you should verify the recipe at the Unit Procedure level.*

Defining an Operator Parameter

To define an operator parameter:

1. In the Recipe Editor, from within a recipe, select the phase you want to associate with the operator parameter.
2. On the Recipe toolbar, click the Step Parameters button. The Step Parameters dialog box appears.
3. From the Origin column of the spreadsheet, select Operator for the parameter you want to modify.

Completing the Recipe Header

To complete the recipe header:

1. In the Recipe Editor, on the toolbar, click the Header Data button. The Header Data dialog box appears.
2. In the Version Number field, enter the recipe's version number.
3. In the Author field, enter your name.

4. If desired, complete the remaining optional fields.

   NOTE: *To rename the recipe and enter a new name in the Procedure Identifier field, use the Save As command. The Save As command restarts the version numbering for the renamed file and assigns a new GUID to the recipe.*

Saving and Verifying Recipes

For information on saving and verifying recipes in the Recipe Editor, refer to the following sections:

- Saving and Verifying a Recipe
- Saving and Verifying All Recipes
- Correcting Errors
- Renaming a Recipe
- Saving a Recipe to a Relational Database
- Converting Recipes

Saving and Verifying Recipes

To save and verify a recipe:

1. In the Recipe Editor, in the tree, click on the recipe that you want to save. All sub-contents of that recipe are included with the save.
2. On the Recipe toolbar, click the Save button.

   NOTE: *The Save button is only enabled if you changed or verified the recipe, or created a new one.*

3. Enter any required electronic signatures. A message box appears asking you to verify the recipe, unless you previously selected this automatic verification check box.
4. Click Verify if an Auto Verify message box appears.
5. Enter any required electronic signatures. The Verification Results window opens.
6. Click OK to close this dialog box.

   TIP: *You can verify all recipes by selecting the Verify Recipes command from the File menu. The toolbar command only verifies the selected recipe and contents.*

Saving and Verifying All Recipes

To verify all recipes:

1. In the Recipe Editor, on the File menu, click Verify Recipes.
2. Enter any required electronic signatures. A message box appears asking if you want to save the changes.
3. Click Yes to continue, or click No to verify but not save the recipe.
4. Repeat steps 2-3 for each recipe that needs to be saved.

   The Verification Results window appears after you save all the recipes.

5. Click OK.

Correcting Errors

To correct an error found after verifying a recipe:

1. In the Recipe Editor, double-click any error in this dialog box to display the associated recipe.
2. Enter any required electronic signatures.
3. Modify the recipe to correct the problem.
4. Re-save the recipe.
5. Enter any required electronic signatures.
6. Re-verify the recipe.
7. Enter any required electronic signatures.

Renaming a Recipe

To rename a recipe:

1. In the Recipe Editor, on the File menu, click Save As. The Save As dialog box appears.
2. Enter the new name of the recipe in the Procedure Identifier field.
3. In the Version Number field, enter the recipe's version number.
4. In the Author field, enter your name.
5. If desired, complete the remaining optional fields.
6. Click Save.
7. Enter any required electronic signatures.

Saving a Recipe to a Relational Database

To save a recipe to a relational database:

1. Start the Proficy Batch Execution WorkSpace.
2. Open the Configuration folder.
3. Select Batch Configuration. The Batch Execution Configuration dialog box appears.
4. Select the Recipe tab.
5. Make sure that a DSN, user name, and password is entered in the ODBC Configuration group box. If it's not, enter it now.
6. Select the SQL option.
7. Save the project.
8. Create the recipes you need with the Recipe Editor.
9. On the File menu, click Save.
10. Enter any required electronic signatures.

    The Recipe Editor automatically saves the recipe to the relational database.

Converting Recipes

To convert recipes in the Recipe Editor:

1. Make sure that the Proficy Batch Execution WorkSpace is open. If it is not, open the WorkSpace and then reopen the Recipe Editor.
2. In the Proficy Batch Execution WorkSpace, in the Batch Execution Configuration dialog box, on the Recipe tab, make sure that you entered a DSN, user name, and password .
3. From the Proficy Batch Execution Recipe Editor, on the File Menu, click Convert Project Storage.
4. Enter any required electronic signatures.

   The Recipe Storage Conversion dialog box appears.
5. Do one of the following steps:
   a. Select the Convert File-Based Project to SQL-Based Project check box and click OK to convert and store the recipes on your hard disk to your relational database.
   b. Select the Convert SQL-Based Project to File Based Project check box and click OK to convert and store the recipes in your relational database to your hard disk.

   The Recipe Editor converts and stores all the recipes in the project.

NOTE: *You can also convert a recipe by changing the Project Storage Type on the Recipe tab of the Batch Execution Configuration dialog box in the Proficy Batch Execution WorkSpace, and then opening the recipe in the Recipe Editor. The Recipe Editor converts the recipes upon opening.*

Releasing a Recipe to Production

To release a recipe to production:

1. In the Recipe Editor, on the toolbar, click the Open button. The Recipe Open dialog box appears.
2. Enter any required electronic signatures.
3. Double-click the recipe you want to release to production. The file opens and its sequential function chart appears.
4. Click the Header Data button on the Recipe toolbar. The Header Data dialog box appears.
5. Select the Released to Production check box.
6. Enter any required electronic signatures.
7. Close the dialog box and save the file. If you do not save the file, Proficy Batch Execution does not release it to production.

   NOTE: *You only need to release procedures to production.*

Printing a Recipe

To print a recipe:

1. In the Recipe Editor, on the toolbar, click the Print button. The Print dialog box appears.
2. Select the printer from the Name field.
3. Click the All button to print the entire recipe. To print a range of pages, select the Pages button and enter the range of pages you want to print.
4. In the Number of Copies field, specify the number of copies to print.
5. Select the Print to File check box to print the recipe to a file.
6. Click OK to print the file.

Creating a Phase Link Group

To create a link group:

1. In the Recipe Editor, open the procedure recipe containing the phases you want to include in the link group.

   NOTE: *You must configure message partners for equipment phases in the Proficy Batch Execution WorkSpace before you create link groups.*

2. On the Link menu, click Link Groups. The Phase Link Group dialog box appears.
3. Select a phase from the Procedural View box.
4. Click the Add button on the Phase Link Group dialog box. The phase is added to the selected column.
5. Repeat steps 3 and 4 to add other phases to the link group.
6. Click OK when you finish adding phases to save the link group.

Modifying a Recipe

For information on modifying a recipe in the Recipe Editor, refer to the following sections:

- Overview: Modifying Recipes
- Defining Equipment Requirements for a Unit Procedure
- Defining Equipment Requirements for an Operation
- Overriding a Unit Procedure's Equipment Requirements
- Modifying the Equipment Requirements
- Modifying Sequential Function Charts
- Modifying Recipe and Step Parameters
- Modifying a Phase Link Group
- Deleting a Phase Link Group

Overview: Modifying Recipes

To modify a recipe:

1. In the Recipe Editor, open the recipe you need to change. Be sure to enter any required electronic signatures.

   The recipe hierarchy appears in the Procedural View box and the recipe's sequential function chart appears in the Construction Area.
2. Modify the recipe's sequential function chart as needed.
3. Modify the recipe's recipe parameters and step parameters as needed.
4. Modify the recipe header to reflect the changes made to the sequential function chart.
5. Modify the equipment requirements as needed.
6. Save and verify the recipe to ensure the recipe is setup correctly.
7. Release the recipe to production.

Defining Equipment Requirements for a Unit Procedure

To define equipment requirements for a unit procedure:

1. In the Recipe Editor, select an operation to highlight it in the SFC.
2. On the Recipe menu, click Equipment Requirements. The Equipment Requirements dialog box appears.
3. If you want to specify a unit class, select the Class Based option, select a unit class from the list box, and select a Default Unit from the list box.
4. If you want to specify a unit instance, select the Instance Based option and select a unit from the list box.
5. In the Unit Procedure Capacity area, enter a value in the Amount field and select a unit of measure from the list box. Optionally, you can enable the Scale Capacity Requirement.
6. Select the appropriate button in the Bind Type area.
7. Select a check box in the Operator Rights area.
8. Click OK.

Defining Equipment Requirements for an Operation

To define equipment requirements for an operation:

1. In the Recipe Editor, select an operation to highlight it in the SFC.
2. On the Recipe menu, click Equipment Requirements. The Override Equipment Requirements dialog box appears.
3. If you want to specify a unit class, select the Class Based option, select a unit class from the list box, and select a default unit from the list box.
4. If you want to specify a unit instance, select the Instance Based option and select a unit from the list box.

Overriding a Unit Procedure's Equipment Requirements

To override a unit procedure's equipment requirements:

1. In the Recipe Editor, on the Recipe menu, at the procedural level, click Equipment Requirements. The Equipment Requirements dialog box appears.
2. Select the Override Unit Procedure Equipment Requirements check box.
3. Select a Default Unit from the list box in the Equipment area.
4. In the Unit Procedure Capacity area, enter a value in the Amount field and select a unit of measure from the list box. Optionally, you can enable the Scale Capacity Requirement.
5. Select the appropriate button in the Bind Type area.
6. Select a check box in the Operator Rights area.
7. Click OK.

   NOTE: *At any time you can click the Fetch Values button to restore the equipment requirements originally configured for this unit procedure.*

Modifying Equipment Requirements

To modify a recipe's equipment requirements:

1. In the Recipe Editor, on the Recipe menu, click Equipment Requirements. The Override Equipment Requirements dialog box appears.

   NOTE: *At the procedural level, you must have the step selected in order to view the Equipment Requirements command in the Recipe menu.*

2. Select the Unit or Unit Class button.
3. Select a specific unit or a unit class required for this recipe.

Modifying Sequential Function Charts

For information on modifying sequential function charts in the Recipe Editor, refer to the following sections:

- Overview: Modifying a Sequential Function Chart
- Adding Steps
- Adding Transitions
- Changing the Font
- Configuring Steps
- Configuring Transitions
- Deleting a Step
- Deleting a Transition
- Deleting Jacobson Links

- Linking Steps and Transitions
- Redefining a Step
- Removing a Link Between a Step and Transition

Overview: Modifying a Sequential Function Chart

To modify a Sequential Function Chart:

1. In the Recipe Editor, delete any steps you no longer need.
2. Delete any transitions you no longer need.
3. Add steps as needed.
4. Add transitions as needed.
5. Configure any new steps.
6. Redefine any existing steps as needed.
7. Configure the transitions as needed.
8. Link the steps and transitions together.

Adding Steps

To add a step to a sequential function chart:

1. In the Recipe Editor, on the SFC toolbar, click the Step tool. The cursor changes to a right angle.
2. Position the cursor where you want the step to appear in the sequential function chart.
3. Click the left mouse button. A new step appears.
4. Repeat this procedure until you have added all the steps you need.

Adding Transitions

To add a transition to a sequential function chart:

1. In the Recipe Editor, on the SFC toolbar, click the Transition tool. The cursor changes to a plus sign.
2. Position the mouse pointer where you want to place the transition in the sequential function chart.
3. Click the left mouse button. The transition appears with its condition set to TRUE.
4. Make your step changes:

Changing the Font

To change the font of all steps in a sequential function chart:

1. In the Recipe Editor, on the Format menu, click Font. The Font dialog box appears.
2. Select the font you want to use from the Font list box.

3. Select the font style from the Font Style list box.
4. Select the font size from the Font Size list box.
5. Click OK to save your changes.

Configuring Steps

To configure a step in a sequential function chart:

1. In the Recipe Editor, on the SFC toolbar, click the Selection tool.
2. Double-click the step you want to configure. A list of the available phases, operations, or unit procedures appears, depending on the level of recipe you are creating.
3. Select a phase, operation, or unit procedure from the list.
4. Click OK. The text in the step changes to reflect your selection.

Configuring Transitions

To configure a transition:

1. In the Recipe Editor, in a recipe, double-click the transition you want to configure. The Transition Expression Builder dialog box appears.
2. Press Delete to clear the current condition and enter a new condition.
3. Use the row of buttons in the middle of the dialog box to add operators to a condition.
4. Use the list boxes at the bottom of the dialog box to select the items you want to add from a list. When using these list boxes:
    - You select an item from the left list box.
    - The categories of the selected item appear in the middle list box.
    - The items in the selected category appear in the right list box.
    - You can add an item to the condition at the top of the dialog box by double-clicking the item from the middle or right list box.
    - If you see a plus sign next to a folder in the left list box, there are items inside the folder. To open the folder, click the plus sign.

Deleting a Step

To delete a step from a sequential function chart:

1. In the Recipe Editor, select the step you want to delete with the Selection tool.
2. Press Delete. The Recipe Editor removes the selected step from the sequential function chart.

Deleting a Transition

To delete a transition:

1. In the Recipe Editor, select the transition you want to delete with the Selection tool.

2. Press Delete. The Recipe Editor removes the selected transition from the sequential function chart.

Deleting Jacobson Links

To delete Jacobson Links:

1. In the Recipe Editor, on the toolbar, click the Remove Jacobson Link tool button.
2. Position the mouse pointer over the origin step.
3. Click and hold the left mouse button.
4. Drag the mouse from the origin step to the destination step.
5. Position the mouse pointer over the destination step and release the mouse button. The red, curved line (Jacobson Link) connecting the origin and the destination steps is

Linking Steps and Transitions

To link a step to a transition:

1. In the Recipe Editor, on the SFC toolbar, click the Link tool. The cursor changes to a link pointer.
2. Position the mouse pointer over the step you want to link.
3. Click and hold the left mouse button.
4. Drag the mouse from the step to the transition.
5. Position the mouse pointer over the transition and release the mouse button. A line appears connecting the step and transition.
6. Repeat steps 1 through 5 alternating between linking steps to transitions and transitions to steps until the entire sequential function chart is complete.

Redefining a Step

To redefine a step in a sequential function chart:

1. In the Recipe Editor, select the step you want to change with the Selection tool.
2. On the Recipe toolbar, click the Redefine Step button. The Select Phase dialog box appears.
3. Select the new unit procedure, operation, or phase you want to use.

Removing a Link Between a Step and Transition

To remove a link between a step and transition:

1. In the Recipe Editor, on the SFC toolbar, select the Remove Link tool.
2. Select a step in the recipe.
3. Click and hold the left mouse button.
4. Drag the mouse to the transition preceding or following the selected step. The Recipe Editor removes the link.

Modifying Recipe and Step Parameters

For information on modifying recipe and step parameters in the Recipe Editor, refer to the following sections:

- Overview: Working with Recipe Parameters and Step Parameters
- Overview: Modifying Recipe Parameters and Step Parameters
- Modifying a Recipe Parameter
- Defining a Constant Value for a Step Parameter
- Modifying Parameter Values
- Deleting a Recipe Parameter

Overview: Working with Recipe Parameters and Step Parameters

To create a step parameter and optionally defer it to a recipe parameter:

1. In the Recipe Editor, define the recipe parameter.
2. Define a step parameter with a constant value. .
3. Defer the step parameter to the recipe parameter, if necessary.

Overview: Modifying Recipe Parameters and Step Parameters

To modify a recipe or step parameter:

1. In the Recipe Editor, modify the recipe parameter.
2. Set a constant value for a step parameter.
3. Modify the step parameter’s value.

Modifying a Recipe Parameter

To modify a recipe parameter:

1. In the Recipe Editor, open the recipe whose recipe parameter you want to modify.
2. On the Recipe toolbar, click the Recipe Parameters button. The Recipe Parameters dialog box appears.
3. Edit the cells in the spreadsheet for the parameter you want to change.

Modifying a Step Parameter

To modify step parameter information:

1. In the Recipe Editor, from within a recipe, right-click the step with the step parameter you want to edit, and click Step Parameters. The Step Parameters dialog box appears.
2. Select the row that you want to edit by placing your cursor in the Origin field.
3. In the Origin column of the selected row, make a selection from the drop-down list:

- **Value** – Use a constant value for the step parameter.
- **Defer** – Defer the value to the next level. This option is provided for backwards compatibility. When deferring parameters, it is recommended that you use the Defer to Schedule, Defer to Procedure, or Defer to Unit Procedure options, because you do not have to manually defer up each level (as you do when you select the Defer option with multiple levels).
- **Defer to Schedule** – Defer the value to the schedule level. The value is provided when you schedule the recipe.
- **Defer to Procedure** – Defer the value to the procedure level.
- **Defer to Unit Procedure** – Defer the value to the unit procedure level.
- **Operator** – Allow the operator to provide the value.

4. In the Value field, enter a value for the parameter. For deferrals, the value you enter here is used to pre-populate the value field at the next level up.
5. Optionally, in the Low field, enter the new low EGU value. This field is available only if the parameter is a Real or Integer type.
6. Optionally, in the High field, enter the new high EGU value. This field is available only if the parameter is a Real or Integer type.

   NOTE: *The low and high level ranges for the step parameters can only be reduced from the phase parameter ranges, not expanded.*

7. If you want to display this parameter and its engineering units (EGU) in the SFC, select the Display check box. If step parameter is a deferred parameter, the SFC step displays the deferred parameter name.
8. Click OK.
9. Verify the recipe to update the step and recipe parameters, and then save it.

   IMPORTANT: *It is very important that you verify the recipe so that your step and recipe parameters are updated. Recipes will need to be verified to the highest level that you chose to defer. For example, if you chose to Defer to Procedure, you should verify the recipes at the Procedure level so the automatic propagation of the parameters will take place. If you select to Defer to Unit Procedure, you should verify the recipe at the Unit Procedure level.*

Modifying Parameter Values

To modify a parameter value:

1. In the Recipe Editor, select the step associated with the parameter you want to modify.
2. Click the Step Parameters button on the Recipe toolbar. The Step Parameters dialog box appears.

Deleting a Recipe Parameter

To delete a recipe parameter:

1. In the Recipe Editor, open the recipe with a recipe parameter you want to delete.
2. On the Recipe toolbar, click the Recipe Parameters button. The Recipe Parameters dialog box appears.

3. Select the entire row of the parameter you want to delete and press Delete. The Recipe Editor removes the selected parameter from the spreadsheet.

   NOTE: *You cannot delete a parameter if a step parameter is deferred to it. You must change the definition of the step parameter before you can delete the recipe parameter.*

Modifying a Phase Link Group

To modify a phase link group:

1. In the Recipe Editor, open the procedure recipe containing the phases you want to modify.
2. On the Link menu, click Link Groups. The Phase Link Group dialog box appears.
3. Select a phase and click the Delete button to remove it from the link group.
4. Repeat step 3 until you have removed all the phases you want.
5. Select a phase from the Procedural View box.
6. Click the Add button on the Phase Link Group dialog box to add a phase to the link group.
7. Repeat steps 5 and 6 to add the other phases you want to include in the link group.

Delete a Phase Link Group

To delete a phase link group:

1. In the Recipe Editor, on the Link menu, click Link Groups. The Phase Link Group dialog box appears.
2. Select a phase referenced in the link group you want to delete and click the Delete button. The Recipe Editor removes the selected phase from the link group.
3. Repeat step 2 for each phase referenced in the link group.
4. Click OK to save your changes.

Deleting Recipes

For information on deleting recipes in the Recipe Editor, refer to the following sections:

- Deleting a Recipe
- Rebuilding the Recipe Directory

Deleting a Recipe

To delete a recipe:

1. In the Recipe Editor, on the File menu, click Remove Recipe. The Remove Recipe dialog box appears.
2. Double-click the name of the recipe you want to delete from the Remove Recipe dialog box. A message box appears.

3. Click the Yes button to delete the selected item or click the No button to return to the Remove Recipe dialog box.
4. Enter any required electronic signatures.

   NOTE: *You cannot delete an open recipe.*

Rebuilding the Recipe Directory

To rebuild the recipe directory:

1. In the Recipe Editor, on the File menu, click Rebuild Recipe Directory.
2. Enter any required electronic signatures.

   If there are any files that need saving, a message box appears.

3. Click Yes to continue and enter any required electronic signatures.

   The Recipe Editor adds each recipe file in your project to the recipe directory. When all the files are added, a message box appears asking you to verify the recipes.

4. If you want to Verify the recipe, click Yes to verify the recipe.
5. Enter any required electronic signatures. The Verification Results dialog box appears.
6. When prompted, click the OK to return to the Recipe Editor.

Working with Recipes

For information on working with recipes in the Recipe Editor and WorkInstruction Editor, refer to the following sections:

- Viewing Audit Versioning Information
- Identifying a report as a key process report
- Exporting a Recipe as an XML File
- Browsing for an EIB
- Assigning an EIB to a Step

Viewing Audit Versioning Information

To view the audit versioning information for the current recipe:

1. In the Recipe Editor, on the Recipe menu, click Audit Information. The Audit Information dialog box appears.
2. Examine each of the fields.

   The fields displayed in the Audit Information dialog box include the audit version number, GUID, and electronic signature information for the user(s) who authorized the Proficy Batch Execution Recipe Editor to save the current version of the recipe.

NOTE: *Information only appears in the electronic signature fields if you enabled auditing and configured signature requirements in the Proficy Batch Execution WorkSpace, and then saved or verified the recipe with the required electronic signatures in the Proficy Batch Execution Recipe Editor. The area model name, GUID, and audit version number always appear.*

Identifying a report as a key process report

To identify a phase report as a key process report:

1. In the Recipe Editor, in the recipe, select the step.
2. Right-click the step and select Key Process Reports, or on the Step menu click Key Process Reports. The Key Process Reports dialog box appears.

   NOTE: *The Key Process Report dialog box is only available from phase level steps.*

3. For each report that you want to make a key process report, select the corresponding check box.
4. Click OK.

Exporting a recipe as an XML file

To export a recipe as an XML file:

1. In the Recipe Editor, open the recipe step (procedure, unit procedure, or operation) in the Recipe Editor, or select it in the tree view.

   NOTE: *Be aware that only the selected step is exported. Steps under the selected step are not exported.*

2. On the File menu, click Export to XML File. The Export Recipe dialog box appears.
3. Leave the defaults, or browse to the folder where you want to save the XML file.
4. Click Save.

Browsing for an EIB

To browse for an EIB:

1. In the Recipe Editor, select a step.
2. On the Step menu, click EWI. The Electronic Work instructions dialog box appears
3. From the Electronic Work Instructions dialog box, click the Browse button. The Select EIB Filename dialog box appears.
4. Select an EIB from the list.

   NOTE: *If no EIBs display in the list box, you need to set up your SourceSafe database connection from the EWI tab on the Batch Execution Configuration dialog box. You need to close the Recipe Editor and launch the Proficy Batch Execution WorkSpace to define these settings.*

5. Click OK.

Assigning an EIB to a Step

To assign an EIB to a step:

1. In the Recipe Editor, select the step.

   NOTE: *These steps assume that you have already set up your EIBs through the WorkInstruction Editor. Click the WorkInstruction Editor button on the Recipe toolbar to access this application.*

2. On the Step menu, click EWI. The Electronic Work instructions dialog box appears.

   NOTE: *The EWI command is only available in the Step menu at the phase level.*

3. Click New.
4. Enter a file name of an EIB in the EIB FileName field or click on the Browse button to search for an EIB file.
5. Enter a version number for the EIB. Select Use Latest Version if you always want to use the most recent EIB version at run time, instead of indicating a specific version to use.
6. Repeat steps 3-5 for each EIB that you want to add to the step.
7. Click OK when you are finished.

Using the Tabular Recipe Editor

For information on using the Tabular Recipe Editor, refer to the following sections:

- Advanced Functions
- Building Recipes
- Formatting the Spreadsheet
- Toggling the Status Bar
- Toggling the Toolbar

Toggling the Status Bar in the Tabular Recipe Editor

To toggle the status bar in the Tabular Recipe Editor:

In the Tabular Recipe Editor, on the View menu, click Status Bar.

Toggling the Toolbar in the Tabular Recipe Editor

To toggle the toolbar in the Tabular Recipe Editor:

In the Tabular Recipe Editor, on the View menu, click Toolbar.

Building Recipes

For information on building recipes in the Tabular Recipe Editor, refer to the following sections:

- Inserting a Recipe Step Anywhere within a Recipe in the Tabular Recipe Editor
- Inserting a Recipe Step at the End of a Recipe in the Tabular Recipe Editor
- Rearranging Recipe Steps in the Tabular Recipe Editor
- Saving a recipe in the Tabular Recipe Editor
- Completing the Recipe Header Information in the Tabular Recipe Editor
- Adding a Sequence Number to a Parameter in an Equipment Phase Class
- Adding Sequence Number Parameters to Tabular Recipes
- Searching and Replacing in Tabular Recipe Editor
- Verifying a Recipe in the Tabular Recipe Editor
- Viewing the Revision History of a Recipe in the Tabular Recipe

Inserting a Recipe Step Anywhere within a Recipe in the Tabular Recipe Editor

To insert a recipe step anywhere within a recipe:

1. In the Tabular Recipe Editor, click anywhere in the spreadsheet.
2. On the Edit menu, click Insert Recipe Step. A step is inserted above the selected row, if a row is selected.

Inserting a Recipe Step at the End of a Recipe in the Tabular Recipe Editor

To insert a recipe step at the end of a recipe:

1. In the Tabular Recipe Editor, on the Edit menu, click Append Recipe Step. A step is inserted at the end of the recipe.

Rearranging Recipe Steps in the Tabular Recipe Editor

To rearrange recipe steps in the Tabular Recipe Editor:

1. In the Tabular Recipe Editor, place your cursor anywhere in the row that you want to move, or select multiple rows by clicking and dragging the cursor on the first gray column, selecting the rows that you want to move. You can also use the keyboard to select multiple rows by clicking on the gray column and pressing the Shift key and the Up or Down keys.
2. Use the up and down arrows in the toolbar to move the row or rows.
3. On the Edit menu, click Renumber Steps, or use the Renumber Steps button on the toolbar.
4. Save your recipe.

Saving a Recipe in the Tabular Recipe Editor

To save a recipe in the Tabular Recipe Editor:

1. In the Tabular Recipe Editor, on the Recipe menu, click Save. The first time you save the recipe, the Save As dialog box appears.
2. Enter a name for the recipe, or leave the default, and click Save.

Completing the Recipe Header Information in the Tabular Recipe Editor

To complete the recipe header in the Tabular Recipe Editor:

1. In the Tabular Recipe Editor, on the Recipe menu, click View Header. The Header Data dialog box appears.
2. In the Version Number field, enter the recipe's version number.
3. In the Author field, enter your name.
4. If desired, complete the remaining optional fields.

Adding Sequence Number Parameters to Tabular Recipes

To add sequence number parameters to tabular recipes:

> ***TIP:*** *If you changed the sequence of your recipe recently, you should renumber the recipe steps before you update or validate the step parameter value. This will make errors easier to find in the Validation Results area of the Parameter Name Settings dialog box. To do this, on the Edit menu, click Renumber Steps.*

1. In the Tabular Recipe Editor, on the Recipe menu, click Configure Parameter Names. The Parameter Name Settings dialog box appears.
2. In the Step Sequence field, leave the default parameter name of SEQUENCE, or enter a new one.

   NOTE: *The name in this field must also be configured as a phase parameter for each equipment phase class in the Batch Execution Equipment Editor, in the Edit Equipment Phase Class dialog box. If it is not, you need to close the Recipe Editor and open the Batch Execution Equipment Editor and do so. For instance, if you leave the default of SEQUENCE, you must add a parameter named SEQUENCE to each phase class in the Equipment Editor. After you add it, you can open the Tabular Recipe Editor again and continue.*
3. Click the Update Current Document Parameter Values button to update the recipe's Step Sequence Number Parameter (as identified by the name entered in step 2) with the step sequence number.
4. Click the Validate Parameter Names in Current Document button to scan the recipe steps for the parameter you entered in step 2. The Tabular Recipe Editor will report back all steps that do not include this parameter.
5. If the log reports phases that do not include this parameter, you may want to add this parameter to those phases in the Batch Execution Equipment Editor. Refer to the Adding a Sequence Number to a Parameter in an Equipment Phase Class section for steps.

Adding a Sequence Number to a Parameter in an Equipment Phase Class

To add a sequence number parameter to an equipment phase class:

1. In the Equipment Editor, in the Create or Edit Equipment Phase Class dialog box, select the Parameters tab.
2. Click the Add button. The Edit Phase Parameter dialog box appears.
3. In the Name field, enter the sequence number's parameter name that you defined in the Tabular Recipe Editor. For example: SEQUENCE.
4. In the Type drop-down list, select Integer.
5. Leave the rest of the fields set to the defaults, and click OK.

Searching and Replacing in the Tabular Recipe Editor

To search and replace a unit in a Tabular Recipe Editor spreadsheet:

1. In the Tabular Recipe Editor, select the row or rows that you want to search, or wait and specify all rows in the Search and Replace dialog box.
2. On the Edit menu, click Search and Replace Phases and Units. The Search and Replace dialog box appears.
3. If you want to search the entire recipe, you can now select the Search and Replace Entire Recipe check box.
4. In the Search and Replace Unit area, in the Find Which Unit drop-down list, select a unit.
5. In the Replace With Unit drop-down list, select a unit.
6. Click Replace All.

To search and replace a phase within a specific unit:

1. In the Tabular Recipe Editor, select the row or rows that you want to search, or wait and specify all rows in the Search and Replace dialog box.
2. On the Edit menu, click Search and Replace Phases and Units. The Search and Replace dialog box appears.
3. If you want to search the entire recipe, you can now select the Search and Replace Entire Recipe check box.
4. In the Search and Replace Phase area, in the Find Which Unit drop-down list, select a unit.
5. In the Search and Replace Phase area, in the Find Which Phase drop-down list, select a phase.
6. In the Replace With Phase drop-down list, select a phase.
7. Click Replace All.

Verifying a Recipe in the Tabular Recipe Editor

To verify a recipe in the Tabular Recipe Editor:

1. In the Tabular Recipe Editor, on the Recipe menu, click Verify Recipe. A message box appears when the verification is complete.
2. Click OK to acknowledge the message.

Viewing the Revision History of a Recipe in the Tabular Recipe Editor

To view the revision history of a recipe open in the Tabular Recipe Editor:

1. In the Tabular Recipe Editor, on the Recipe menu, click View Revision History. The Revision History dialog box appears.
2. Review the revisions listed in the grid. Every time you use Save or Save As for the selected recipe, the revision is recorded in this log. The revision information is saved as part of the tabular recipe file.
3. If you want to suppress the message box that appears when you select Save or Save As, select the Turn Off Revision Log Prompt check box. By default, this check box is selected. When selected, all Save and Save As actions are logged under the Anonymous user.

   Clear this check box to require the user to enter a user name every time there is a Save or Save As action performed for this recipe.

4. Click OK to close this dialog box.

Formatting the Spreadsheet

For information on formatting the spreadsheet in the Tabular Recipe Editor, refer to the following sections:

- Coloring a row or rows in a spreadsheet
- Bolding the text of a row or rows in a spreadsheet
- Deleting a row or rows in a spreadsheet

Coloring a Row or Rows in a Spreadsheet in the Tabular Recipe Editor

To color a row or rows in a spreadsheet:

1. In the Tabular Recipe Editor, place your cursor anywhere in the row that you want to color, or select multiple rows (by clicking and dragging) that you want to move.
2. On the Edit menu, click Pick Color. The Color dialog box appears.
3. Select a color and click OK.

   NOTE: *You can create a custom color by clicking the Define Custom Colors button.*

Bolding the Text in a Row or Rows in a Spreadsheet in the Tabular Recipe Editor

To bold the text of a row or rows in a spreadsheet:

1. In the Tabular Recipe Editor, place your cursor anywhere in the row that you want to color, or select multiple rows (by clicking and dragging) that you want to move.
2. On the Edit menu, click Toggle Bold Font.

   NOTE: *To remove the bold, select the row or rows and on the Edit menu, click Toggle Bold Font again.*

Deleting a Row or Rows in a Spreadsheet in the Tabular Recipe Editor

To delete a row or rows in a spreadsheet:

1. In the Tabular Recipe Editor, place your cursor anywhere in the row that you want to delete, or select multiple rows (by clicking and dragging) that you want to move.
2. Click the Delete button on the toolbar.

Performing Advanced Functions

For information on advanced functions in the Recipe Editor, refer to the following sections:

- Transforming a recipe to an SFC for use in the original Recipe Editor
- Importing a .txt File into the Tabular Recipe Editor
- Exporting a .txt File into the Tabular Recipe Editor
- Generating a Report in the Tabular Recipe Editor

Transforming a Recipe to an SFC for Use in the Original Recipe Editor

To transform a recipe to an SFC for use in the original Recipe Editor:

1. In the Tabular Recipe Editor, open the recipe.
2. On the Recipe menu, click Verify Recipe. A message box appears after the verification process.

   IMPORTANT: *The recipe must be verified before you can transform the recipe. If you do not verify the recipe, it will be done when you transform it. Verifying it beforehand helps you identify issues before you perform the transform.*

3. On the File menu, click Transform Recipe to SFC Batch System. A message box appears when the transform is complete.
4. Click OK.

Importing a .txt File into the Tabular Recipe Editor

To import a .txt file into the Tabular Recipe Editor:

1. In the Tabular Recipe Editor, on the File menu, click Import. The Import from Tab Delimited Text File dialog box appears.
2. Select or browse and select the .txt file that you want to import, and click Open. The recipe that you want to import appears in a new recipe in the Tabular Recipe Editor window.
3. On the File menu, click Save As.
4. Enter a name to save the file as, and click Save.

Exporting a .txt File into the Tabular Recipe Editor

To export a .txt file in CSV format from the Tabular Recipe Editor:

1. In the Tabular Recipe Editor, open the recipe and make sure that it is the active window.
2. On the File menu, click Export. The Export to Tab Delimited Text File dialog box appears.
3. In the File Name field, leave the default name or enter a new one.
4. Leave the default folder or browse to another folder. By default, recipes are exported to the project's Recipe folder. For example :C:\Program Files\Proficy\Proficy Batch Execution\Projects\DEMO\RECIPES.
5. Click Save.

Generating a Report in the Tabular Recipe Editor

To generate a report in the Tabular Recipe Editor:

1. In the Tabular Recipe Editor, open the recipe.
2. On the File menu, click Generate Report. The Save Report dialog box appears.
3. In the File Name field, leave the default name or enter a new one.
4. Leave the default folder name or browse to another folder. By default, recipes are saved to the project's Recipe folder. For example: C:\Program Files\Proficy\Proficy Batch Execution\Projects\DEMO\RECIPES\TabularRecipeReports.
5. Click Save. After a few moments, a report window appear.
6. Optionally, click any of the buttons at the top of the report to format or view it differently.

Index

D

E

F

H

I

J

R

S