Understanding
Sequential Control Modules

Introduction

A Sequential Control Module (SCM) is a graphic oriented program that is configured using the Control Builder Program. SCMs are made up of Handlers which are composed of Transitions and Steps, that when combined, form a sequential process. This lesson is designed to explain the concept and use of SCMs in a typical project operation.

Objectives

- Understand the Components of an SCM
- Understand the procedure used to Implement an SCM
- Understand the rules that apply to SCMs

SCM Features

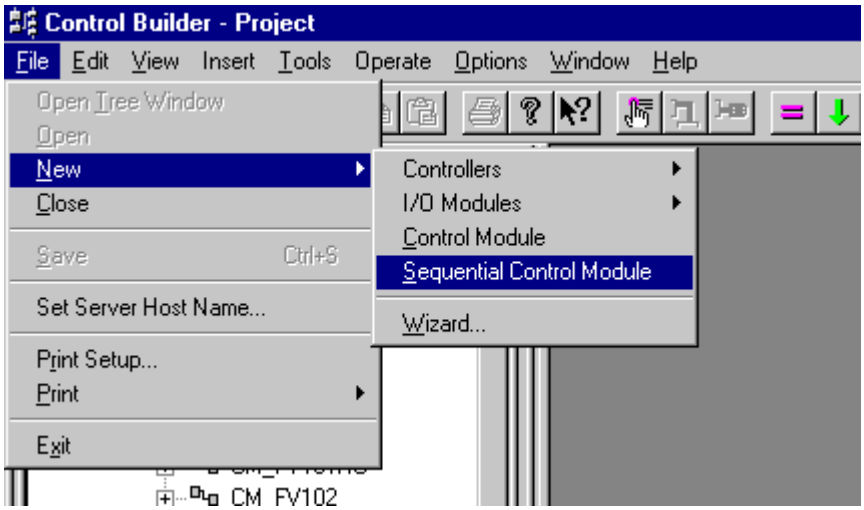
- Abnormal condition handling capability
- Computational capability
- SCM structure relates directly to the physical process
- Programming requirements are limited to modulating, sequential and logical control functions that relate directly to the process
- Statements, which are elements of a SCM, consist of English language descriptions and process related codes that make SCMs easy to learn and use

Creating a Sequential Control Module (SCM)

To create an SCM you can use one of three methods. The methods are similar to the methods used to create Control Modules.

First Method

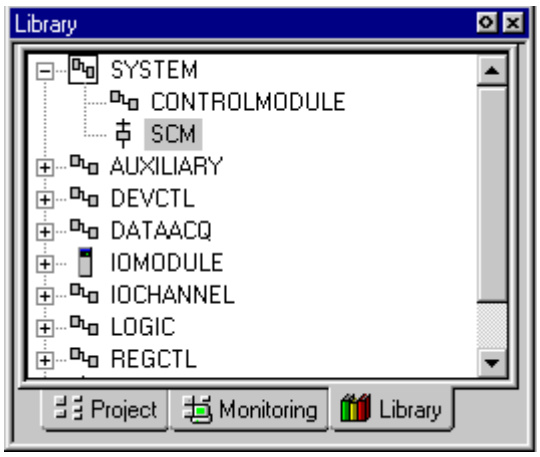
From the dropdown menu Select **File > New > Sequential Control Module**



Creating a Sequential Control Module (SCM) *…continued*

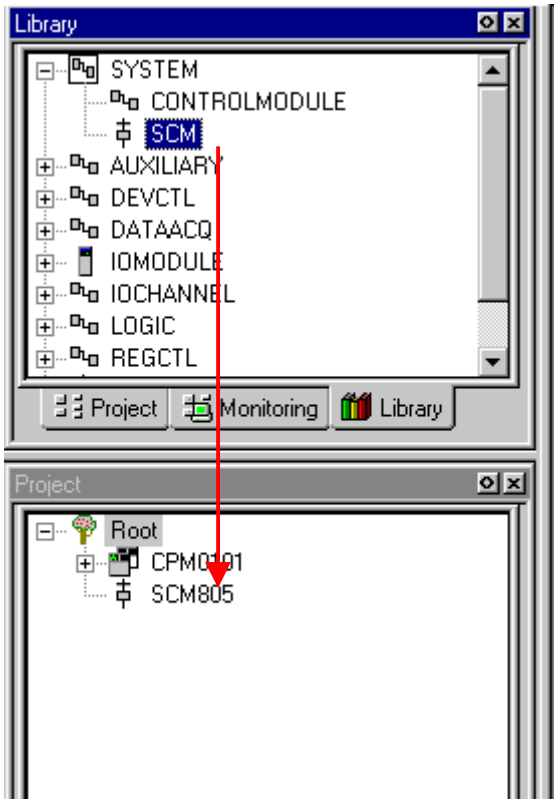
Second Method

From the Library Tab expand the system directory. This will expose the option of creating a Control Module or an SCM, **Double Click on the SCM**



Third Method

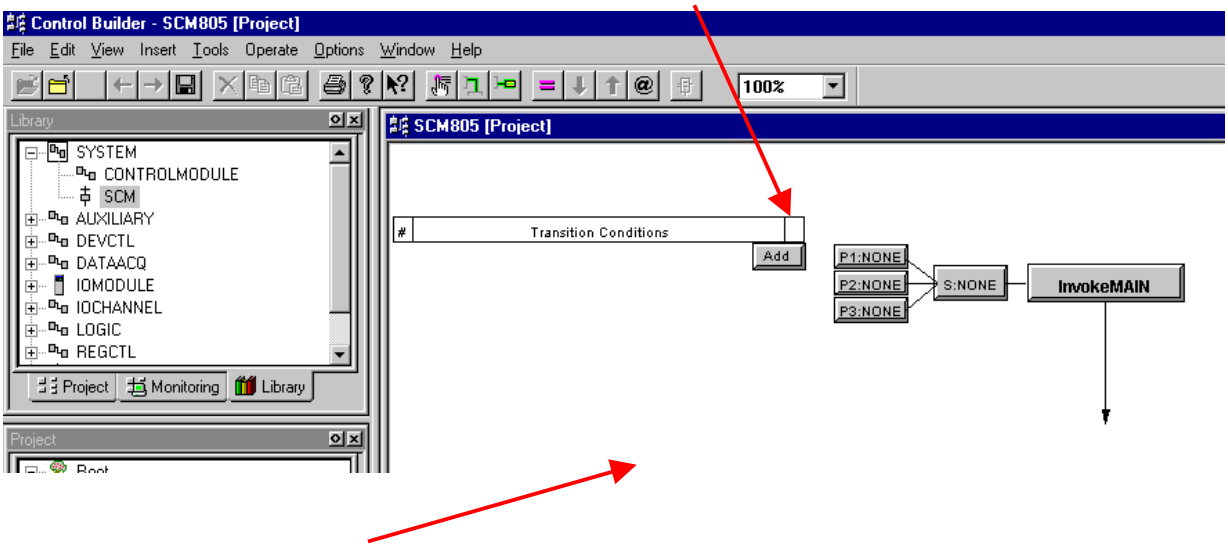
From the Library Tab expand the system directory. Open a second Tree View and select the project tab. Select SCM from the Library tab and drag it over to the Project tree.



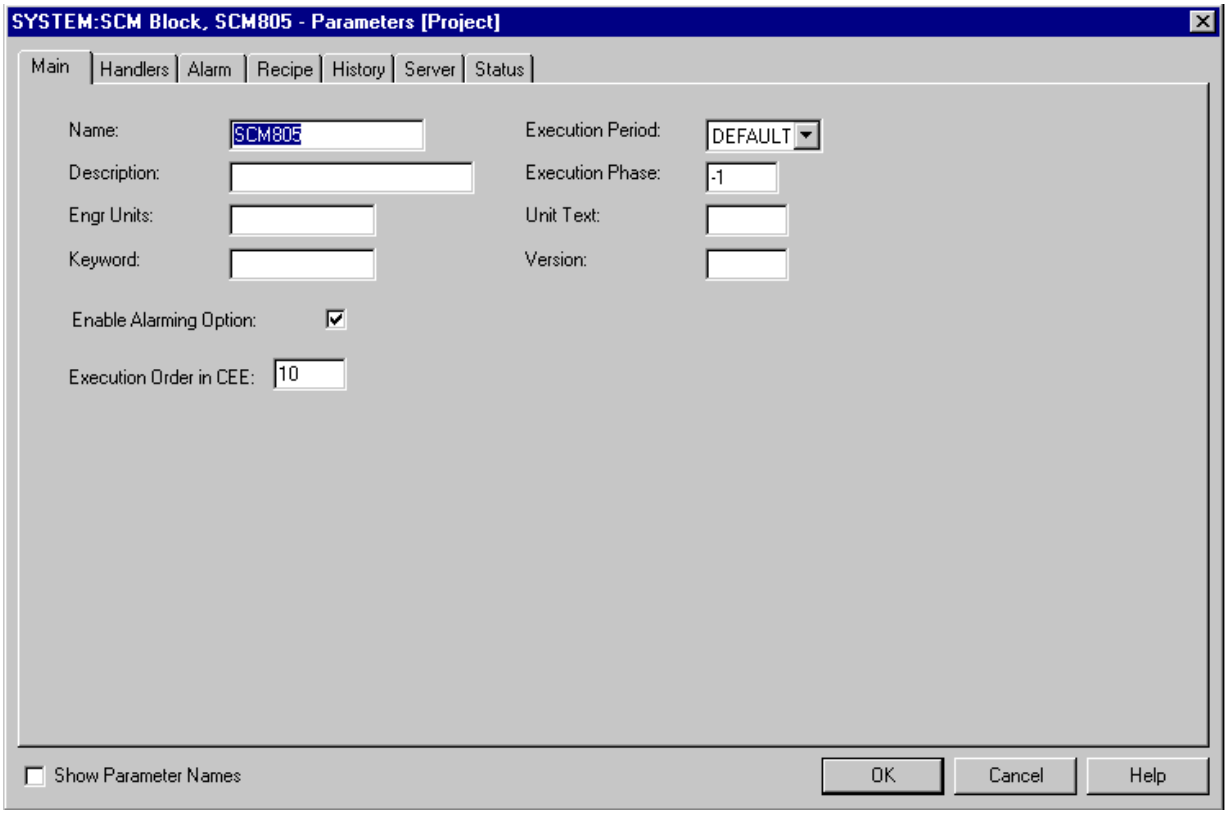
Once the SCM has been created it will appear under the Root Project Tree. The SCM will automatically be given a name (In this case SCM805).

Configuring an SCMs Parameter Configuration Form

After creating an SCM it will appear in the Control Drawing area.



Double clicking on any unoccupied space in the control drawing area will give you access to the Parameter Configuration form.



Configuring the Main Tab

Name: Enter a name that represents the function this SCM will have in your project

(Tank_A_XFER)

Description: Enter a brief description of the SCM's purpose (Tank A Transfer)

Engr Units: Enter the unit of measurement the SCM will be using (GAL, LBS etc.)

Keyword: Enter a short name that will be displayed on the Station SCM Faceplate (A_XFER)

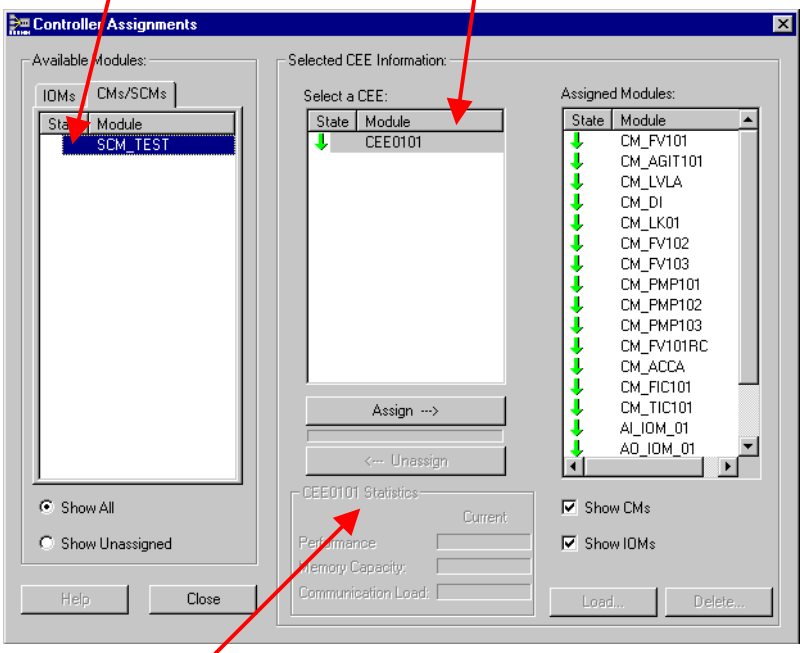
Assigning an SCM to a CEE

Before an SCM can be properly configured it must be assigned to a Control Execution Environment (CEE). A sequence has been established to facilitate the assignment process. Following this sequence should avoid problems during assignment.

Step 1: Ensure the SCM is **not open** in the control drawing area

Step 2: Select the Assign button from the toolbar

Step 3: Select the SCM to be assigned and the CEE you wish to assign it to



Step 4: Click on the Assign Button

Step 5: Click Close

Your SCM will now appear under the CEE you selected in the Project Tab. This will enable you to use parameters from the objects assigned to this CEE.

SCM Structure

An SCM is composed of Handlers. Handlers are composed of Transitions and Steps that are soft wired together.

The internal structure of each handler is in direct relation to the steps of the process being configured. Each Handler **must** start with a Transition, which is then followed by a step, which is then followed by a Transition, and so on. Using this sequence allows the programmer to start tasks in motion with the Step, and then verify they are running (or completed) with a Transition.

Handlers

Types of handlers	Internal	Normal	Abnormal
	(Non configurable)		
Null	**X**		
Edit	**X**		
Main		**X**	
Check		**X**	
Interrupt		**X**	
Restart			**X**
Hold			**X**
Stop			**X**
Abort			**X**

Only one type of Handler may be configured at a time, and only one Handler may be active at any given time.

SCM Structure …*continued*

Internal Handlers

Null Handler

The **Null Handler** is not configurable. The Null Handler is the handler that the system uses when a Check, Interrupt, Hold, Restart, Stop or Abort Handler is commanded yet not configured. For example, when a Main Handler is completed, it returns to Idle State via the Check Handler. If there is no Check Handler configured, the Null Handler is used.

Edit Handler

The **Edit Handler** is not configurable. The Edit handler is executed when the SCM transitions from an Inactive state to an Active state. The Edit Handler completes initialization activities and SCM validation. SCM validation looks for and identifies configuration errors in all Handlers, Steps and Transitions of the SCM. When the SCM is Validated / Idle state, the configuration status of the SCM and its components should be checked by the user before starting / using the SCM.

SCM Structure …*continued*

Normal Handlers

Main Handler

The **Main Handler** of the SCM contains the primary sequential activities of the process. The Main Handler is the core of the SCM.

Check Handler

The SCM executes the **Check Handler** just before entering the IDLE state. On initial entry to the SCM, the Check Handler is executed as soon as the SCM is activated. The SCM also returns to the Check Handler after

- a *RESET* command or when the conditions for the Check Handler are met after the STOP Handler is completed
- the MAIN Handler is completed
- the ABORT Handler is completed

A configured Check Handler could be used to initialize process equipment and/or reset values for a new activity.

Interrupt Handler

The **Interrupt Handler** interrupts the activity of the Main Handler acting like a subroutine of the Main Handler. The Interrupt Handler begins executing when

- the Step's interrupt conditions are met
- the Invoke Transition's conditions are met
- an operator command is given

SCM Structure *…continued*

Abnormal Handlers

Restart Handler

The **Restart Handler** returns the activity to the Main/Interrupt Handler from the Held State. The main handler is restarted at the step designated as the Restart Address.

Stop Handler

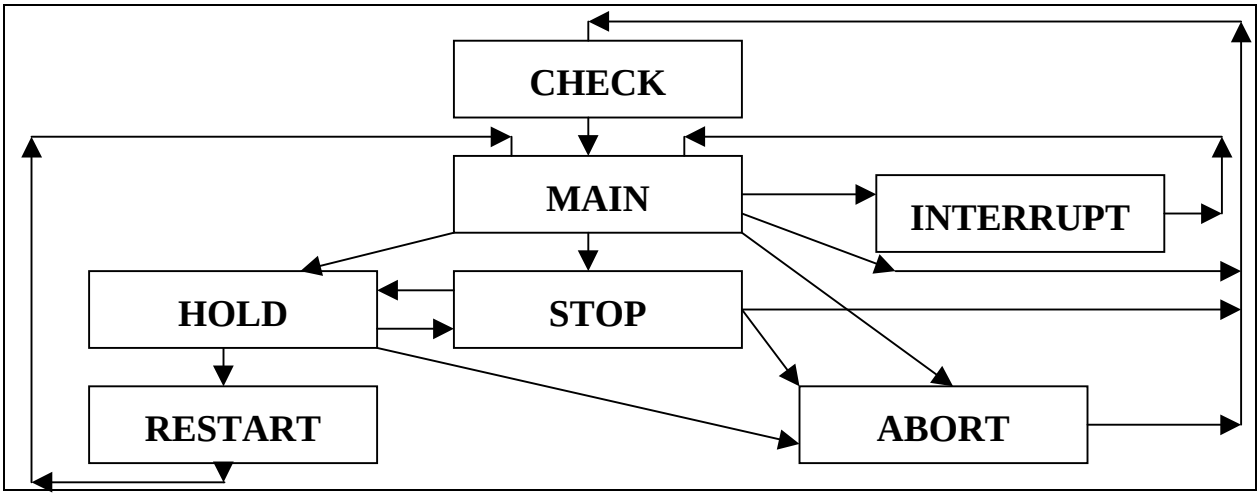
The **Stop Handler** preempts the activity of the Main/Interrupt, Restart or Hold Handlers. The Abort Handler can preempt the Stop Handler. From the Stop Handler you can go automatically to the Abort or Check Handler or by command to the Hold Handler.

Hold Handler

The **Hold Handler** preempts the activity of the Main and/or Interrupt Handlers. The Stop and Abort Handlers can preempt the Hold Handler. From the Hold Handler you can go to the Restart, Stop or Abort Handler.

Abort Handler

The **Abort Handler** preempts the activity of the Main/Interrupt, Restart, Hold or Stop Handlers. The Abort Handler cannot be preempted. From the Abort Handler you can only return to the Check Handler by command or by meeting the conditions of the Check Handler.



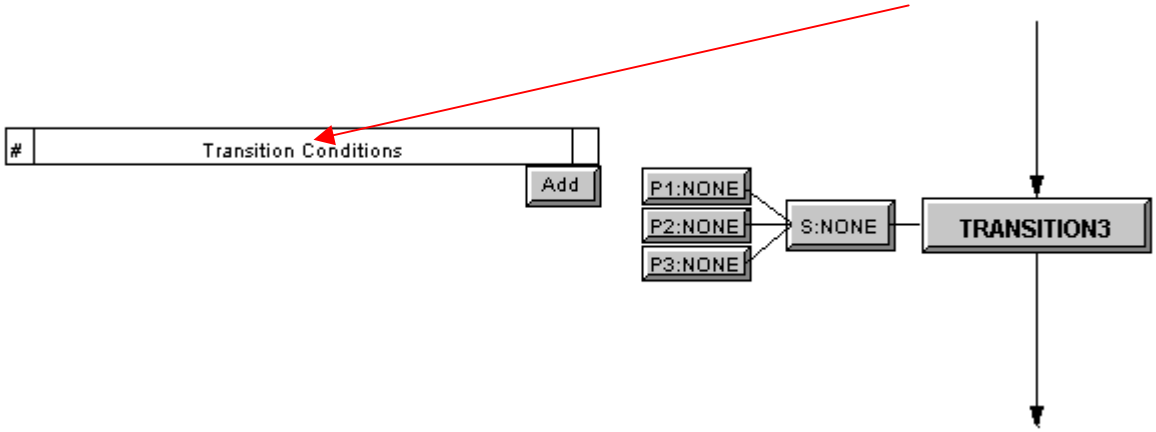
Creating a Transition

A Transition contains one or more conditions with logic gates that must be logically TRUE (=1) before the SCM will continue execution. Up to 10 conditions can be placed on each Transition. Conditions are logically connected using three primary logic gates and one secondary gate. A Transition can be considered to be one big wait statement.

All transitions appear in two views in the SCM

- The Description view (Provides the Description entered in the configuration form)
- The Expression view (Provides the Expression entered in the configuration form)

You can Easily change between the two views by clicking on the condition banner.



Invoke Transition

Every Handler **must** start with a Transition. The first transition of a Handler is automatically added and configured as the Invoke Transition. This can be verified by:

- Selecting the transition
- Pulling down the edit menu
- Verifying "Invoke Transition ON" is checked

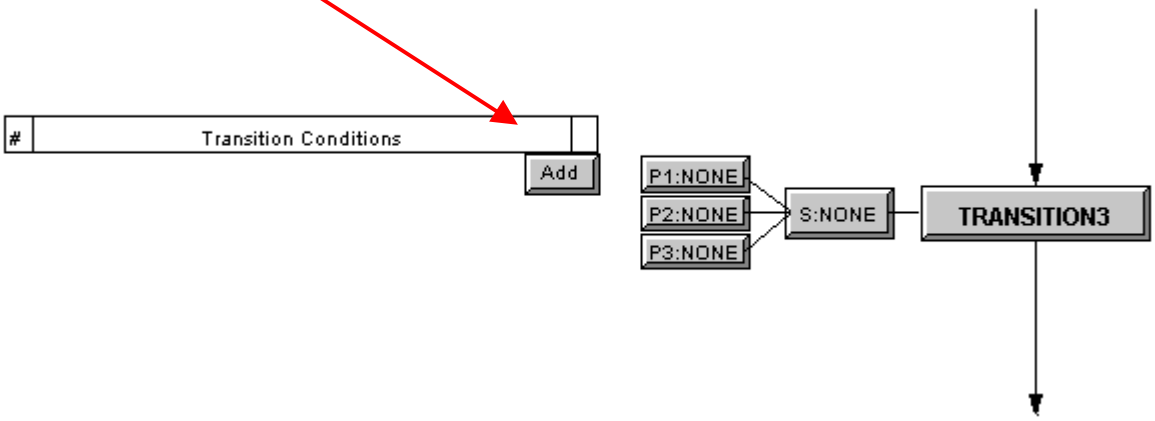
A handler **must** have an Invoke Transition – if it doesn't, the start command will skip all Steps and Transitions of the Handler and go directly to Complete.

If you use the Default Invoke Transition as provided, the Handler requires an operator Start command each time that it is to run. If you want the Handler to start automatically due to process inputs, the Invoke Transition must contain statements that can be evaluated as true or false. If evaluated to be true, the Handler will start.

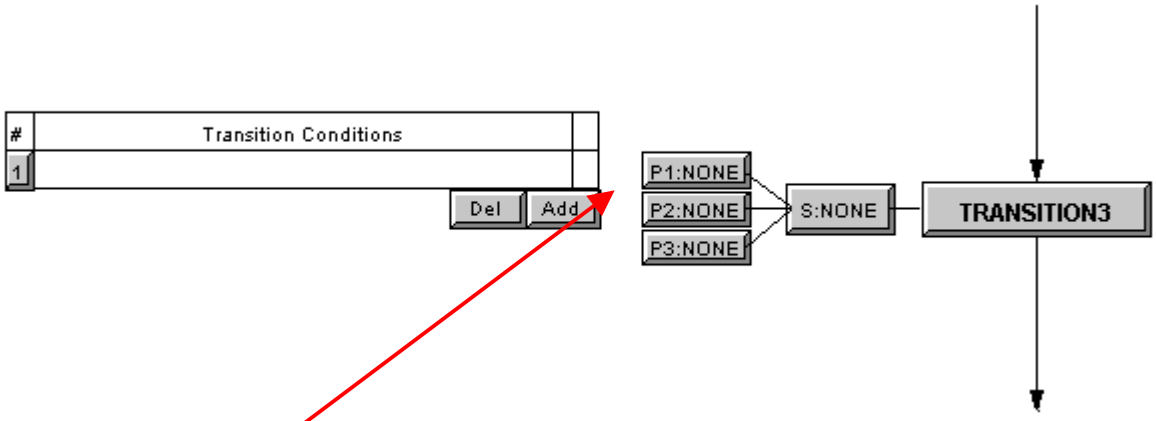
Creating a Transition *…continued*

Using and Configuring Transition

When you add a Transition it does not have any Conditions available. To add conditions you must click on the **Add** button. Every time you click on the **Add** button it will add another Condition.



Once a Condition has been added a new box will appear below the "Transition Condition Description" Box. In addition the Delete **Del** button will appear next to the **Add** button.



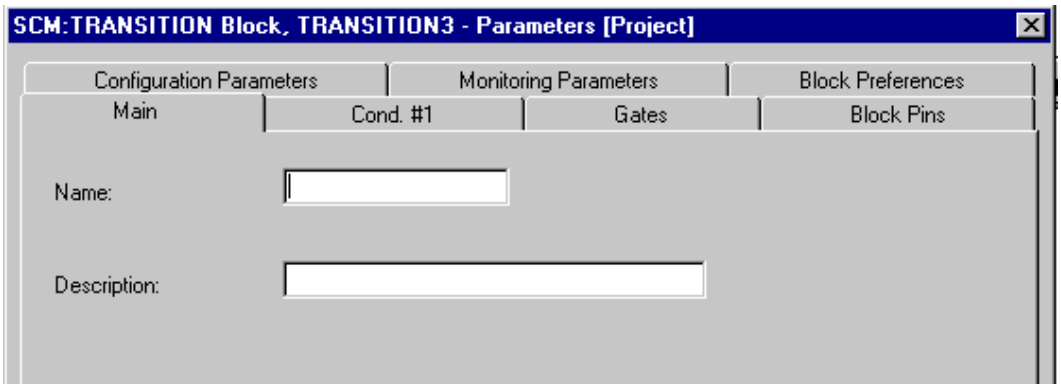
Note: No connections appear between the Conditions and the Gates, they must be manually configured in the Parameter Configuration form.

Creating a Transition …*continued*

Parameter Configuration Form

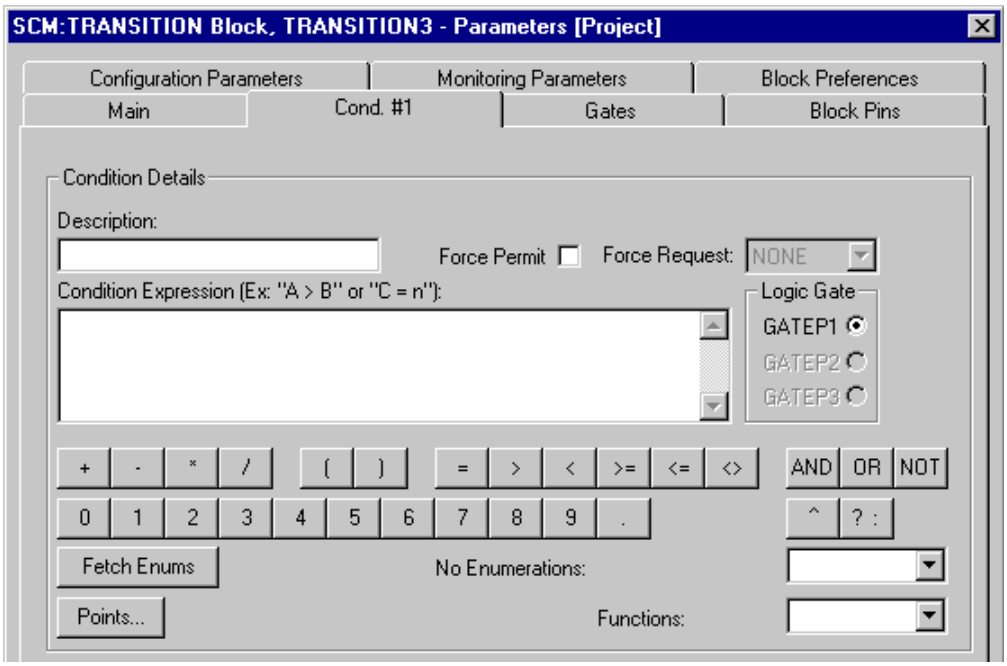
On the Parameter configuration form you will find several tabs such as Main, Cond #1-10 and Gates.

Main Tab The Main Tab is where you enter the Name and Description of the Transition you are configuring



Cond # Tab You will have a Cond # Tab for each condition you add to your Transition. In this Tab you will find

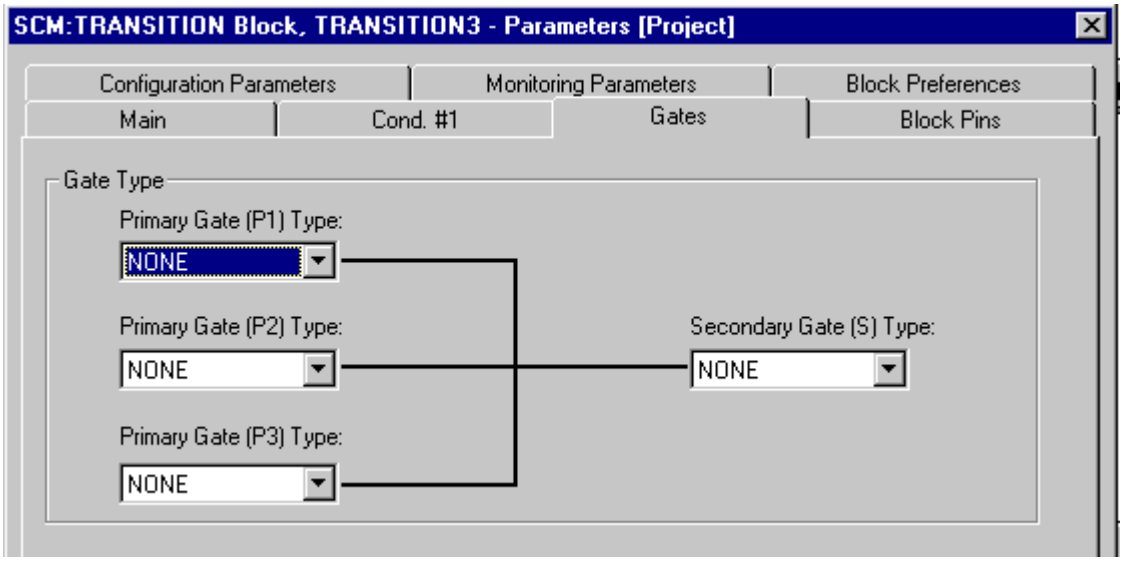
- Description field: Description of Condition you will be Implementing (23 characters maximum)
- Condition Expression field: Logical Expression to execute a function in Control Builder (written in "system" expression syntax)



Creating a Transition *…continued*

Parameter Configuration Form *…continued*

Gates Tab The Gates tab is used to logically join the conditions to each other.



Expressions

Expressions can be as simple as "X = Y" or as complicated as you need them to be. Simple mathematical code is used to accomplish this.

Quantifiers

<	Less than	>	Greater than
<=	Less than or equal to	>=	Greater than or equal to
=	Equal to	<>	Not equal to

Mathematical Operators

+	Addition	*	Multiplication
/	Division	/	Division returning integer result
MOD	Modulus operator	^	Exponent operator

Creating a Transition …*continued*

Expressions …*continued*

Logical Operations

AND	OR	NOT

Single Argument Functions

ABS	Absolute value	ATN	Arc tangent
SIN	Sine of an angle	COS	Cosine of an angle
TAN	Tangent of an angle	Log	Natural logarithm of a number
LN	Log to the base of e	SQR	Square of a number
SQRT	Square root of a number	EXP	Exponential EXP(2.3) = 10
INT	Integer INT(2.3) = 2	ISFIN	Is Finite
ISNAN	Is NAN ISNAN(2.3) = 0 (false) ; ISNAN(Nan) = 1 (true)	RND	Round RND(3.7) = 4
SGN	Sign SGN(-3) = -1; SGN(2.4) = 1; SGN(0) = 0		

Multiple Argument Functions

MIN	Minimum value	MAX	Maximum value
AVG	Average values	MID	Median
MUL	Multiply	SUM	Sum

Creating a Transition *…continued*

Expressions *…continued*

SCM States

0	Nulling	1	Null
2	Inactive	3	Validated
4	Running	5	Complete
6	Checking	7	Idle
8	Interrupting	9	Interrupted
10	Restarting	11	Restarted
12	Holding	13	Held
14	Stopping	15	Stopped
16	Aborting	17	Aborted
18	CommErr (Communication Error)		

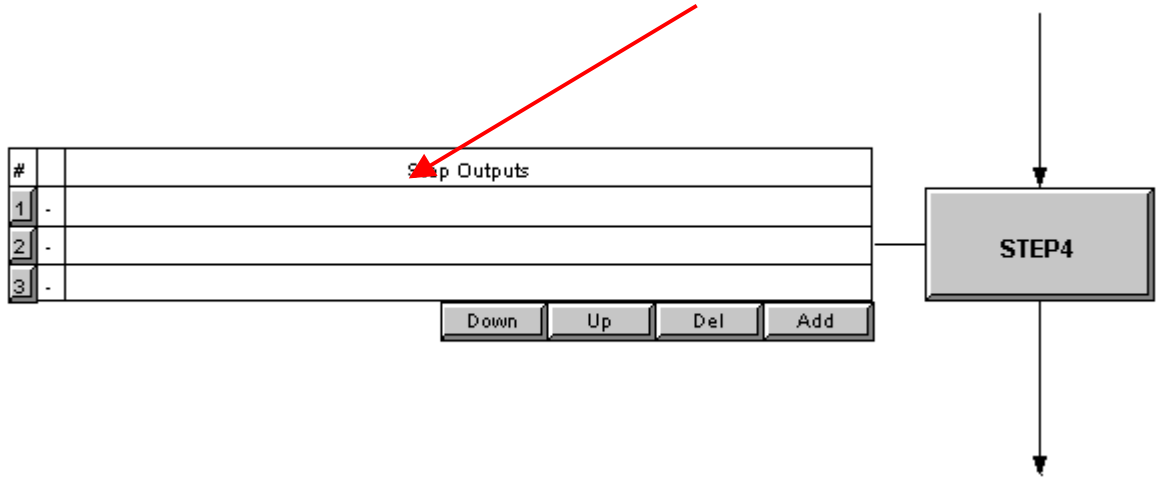
Creating a Step

A Step contains one or more executable output statements. Up to 16 outputs can be part of a single step. If the Step block is too close to surrounding blocks, Control Builder will not permit you to enter more outputs to the Step. To ensure that you will be able to enter all needed output statements, add outputs before adding the next transition.

Steps appear in two views in the SCM

1. The Description view (Provides the Description entered in the configuration form)
2. The Output view (Provides the Output Expression entered in the configuration form)

You can easily change between the two views by clicking on the Step banner.

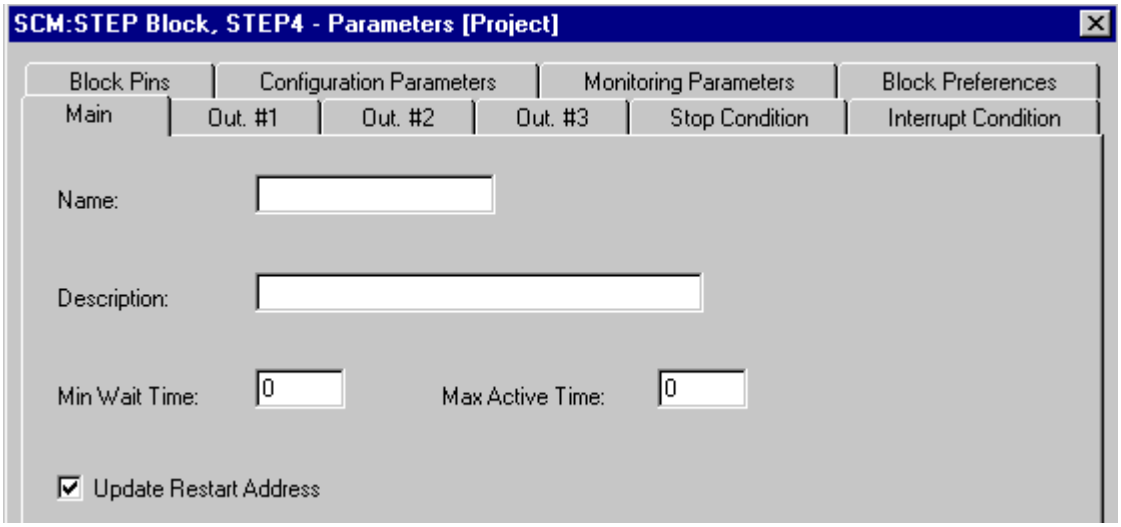


Creating a Step *…continued*

Parameter Configuration Form

On the Parameter configuration form you will find several tabs such as Main, Out #1- 16, Stop Condition and Interrupt Condition.

Main Tab The Main tab of the Step Parameters configuration form allows entry of Name, Description, Minimum Wait Time, Maximum Active Time and Update Restart Address.



Name: The Name entered on this form is used to describe the entire Step. A maximum of 16 characters can be used in the name. It is important to remember that the name being entered will be a part of the code used, therefore it must not contain breaks.

Examples (Turn_Pump_On or SCM_To_Run)

Description: The Description can be up to 39 characters long and can contain breaks.

Examples (Turn Pump On or SCM to Run)

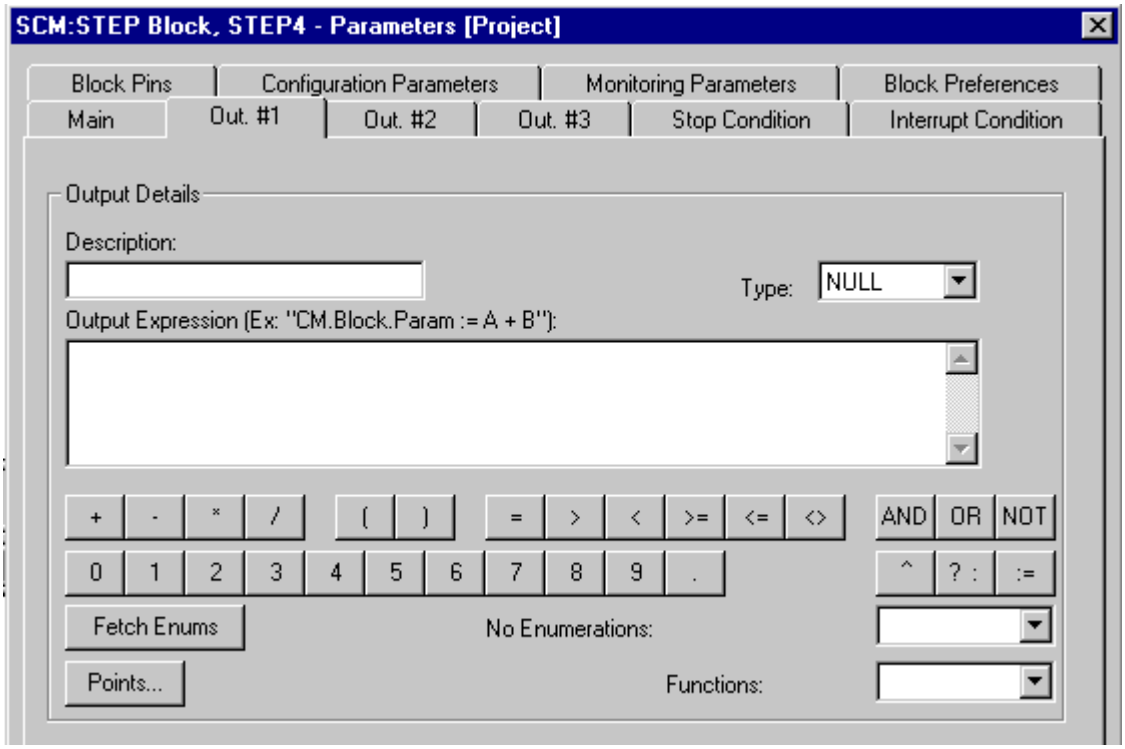
Min Wait Time: The minimum wait time entered here will determine the length of time the Step will wait before proceeding on to the next Transition. The time is entered in cycles. This feature is important when timing is critical such as when many Output Expressions are required and interlocks need time to clear, to avoid interlock alarms.

Max Active Time: This time is set to enable the Step Timeout Alarm. The Alarm will be activated if all the Output Expressions have not completed in the prescribed time. This time is also entered in cycles.

Creating a Step …*continued*

Parameter Configuration Form …*continued*

Out # Tab Each Step can have up to 16 outputs configured with it.



Output Description: The Output Description pertains directly to the Output Expression entered on the same Tab. If Out #1 was going to be used to change the Mode of a Device Control Block to Program the Description might read "FV101 DEVCTL to PGM". Up to 23 characters can be used for the Description.

Output Expression: The Output Expression is the "code" that will be used to perform tasks for you in your project. The Expression can not be entered unless the points being referenced exist in the Project. This would be broken down into

(Point or object to be controlled or set) := (Value or command set)

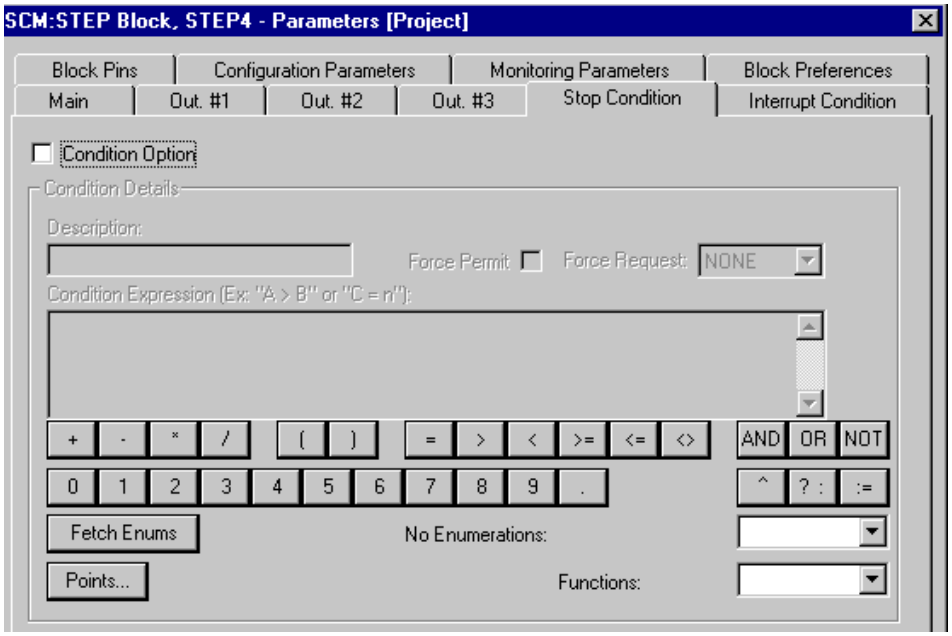
Example: TIC101.PIDA.SP := 40.0 (This would set the CMs (TIC101) PIDA blocks SP to 40)

Creating a Step …*continued*

Parameter Configuration Form …*continued*

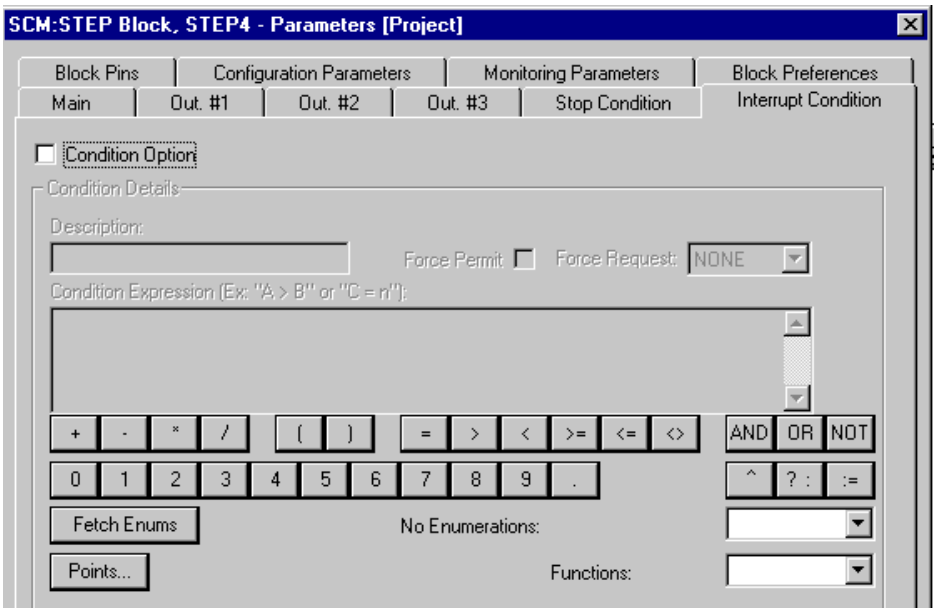
Stop Condition

Each Step can be configured with its own Stop Condition. When the condition is met, if configured, a Stop Handler will run ; or if no Stop Handler, the Main Handler will go to the Stopped State.



Interrupt Condition

Each Step can be configured with its own Interrupt Condition. When the condition is met, if configured, an Interrupt Handler will run ; or if no Interrupt Handler, the Main Handler will alternate between Running and Interrupting until the condition clears.



Creating a Step …*continued*

Expressions

Expressions can be as simple as "X := Y" where:

X = Module.Function_Block.Parameter **:=** = Assignment operator Y = Desired value

or as complicated as you need them to be. Simple mathematical code is used to accomplish this.

Mathematical Operators

+	Addition	*	Multiplication
/	Division	/	Division returning integer result
MOD	Modulus operator	^	Exponent operator

Logical Operations

AND OR NOT

Single Argument Functions

ABS	Absolute value	ATN	Arc tangent
SIN	Sine of an angle	COS	Cosine of an angle
TAN	Tangent of an angle	Log	Natural logarithm of a number
LN	Log to the base of e	SQR	Square of a number
SQRT	Square root of a number	EXP	Exponential EXP(2.3) = 10
INT	Integer INT(2.3) = 2	ISFIN	Is Finite
ISNAN	Is NAN ISNAN(2.3) = 0 (false) ; ISNAN(Nan) = 1 (true)	RND	Round RND(3.7) = 4
SGN	Sign SGN(-3) = -1; SGN(2.4) = 1; SGN(0) = 0		

Multiple Argument Functions

MIN	Minimum value	MAX	Maximum value
AVG	Average values	MID	Median
MUL	Multiply	SUM	Sum

Creating a Step …*continued*

Expressions

SCM Commands

0	None	1	Inactive
2	Start	3	Reset
4	Interrupt	5	Restart
6	Hold	7	Stop
8	Abort	9	Resume
10	Active		

SCM Modes

0	None	1	Auto
2	SemiAuto	3	SingleStep
4	Manual	5	Normal

Mode Attributes (SCM & CM)

0	None	1	Operator
2	Program	3	Normal

Device Control Block States GOP/GPV

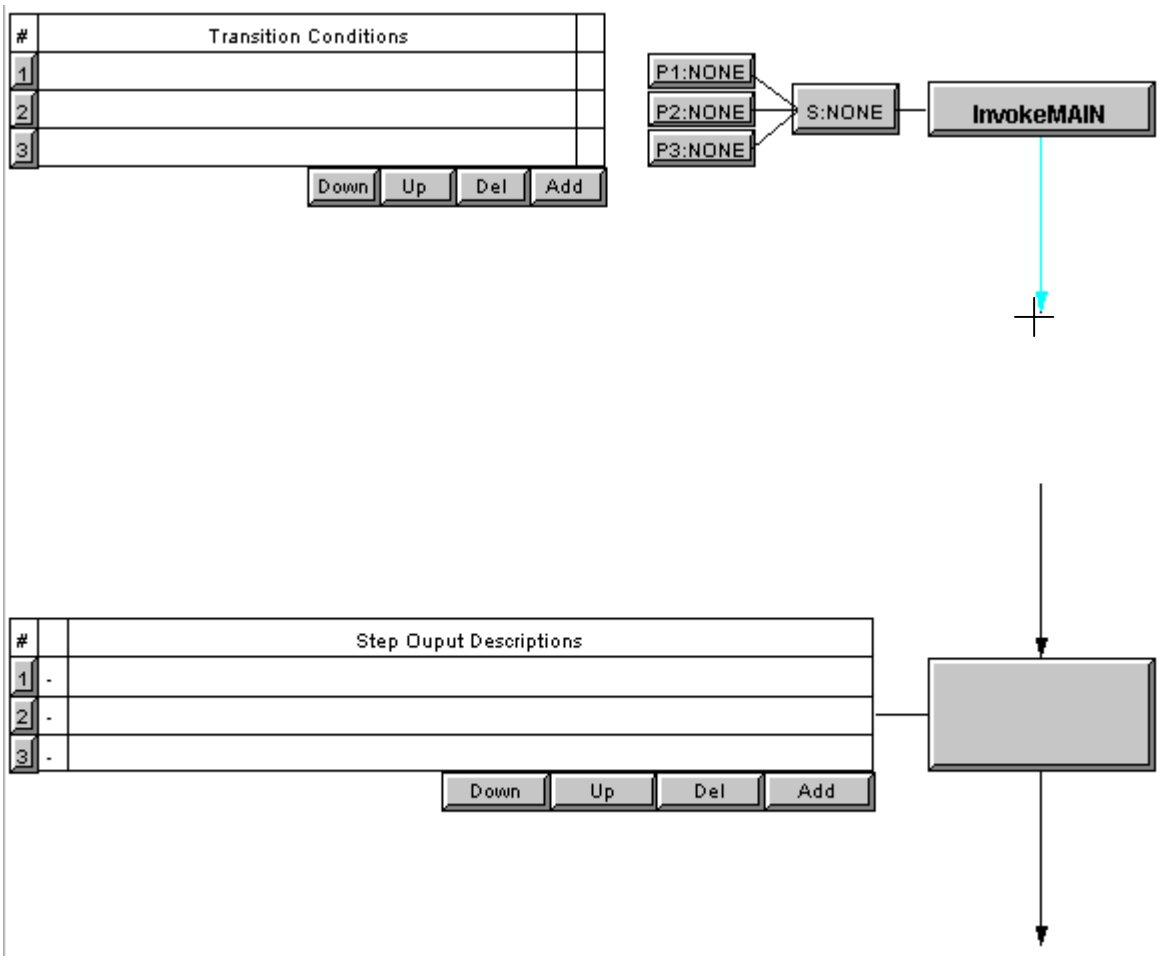
4	S0 (Off/STOP)	5	S1 (ON/START)
6	S2 (REVERSE)		

Wiring Within an SCM

Wiring a SCM is identical to wiring a CM:

- Position the cursor over the Pin of the Step or Transition you wish to connect
- Double Click and the pointer will turn into a Plus Sign +
- Route the wire to the Step / Transition to which you wish to connect
- Double click on the Pin to which you wish to connect

You must follow the sequence of Transition / Step / Transition / Step ……..

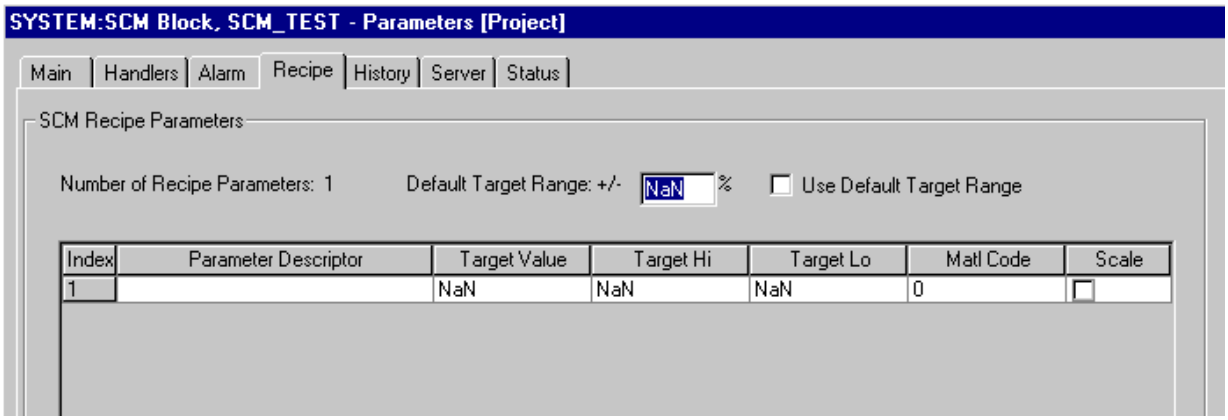


Recipe Parameters

Every SCM can contain up to 50 sets of Recipe Parameters. An SCM allows access to its Recipe Parameters, once it is Active, from the SCM Detail Display. The Recipe Manager option enables point parameters for sets of points to be downloaded with pre-configured working values. The individual point parameters are the recipe "ingredients."

Accessing the Recipe Information

To view or configure the Recipe information in an SCM you must open the parameter configuration form. Once open you will see a tab marked Recipe. Selecting this tab will allow you to configure the different settings contained within the Recipe Parameters



Recipe Parameters *…continued*

- **Parameter Descriptor**

  Description of up to 23 characters.

- **Target Value**

  Specify a real number for recipe parameter target value.

- **Target Hi**

  Specify a real number of the maximum value of the Target Value or use default value, if enabled.

- **Target Lo**

  Specify a real number for the minimum value of the Target Value or use default value, if enabled.

- **Material Code (Matl Code)**

  Specify integer that represents the material ingredient of the recipe parameter.

- **Scale**

  Select whether parameter can be scaled (Yes) or not (No). If not, the RECSCALE{1..50} parameter is set to off.

- **Use Default Target Range**

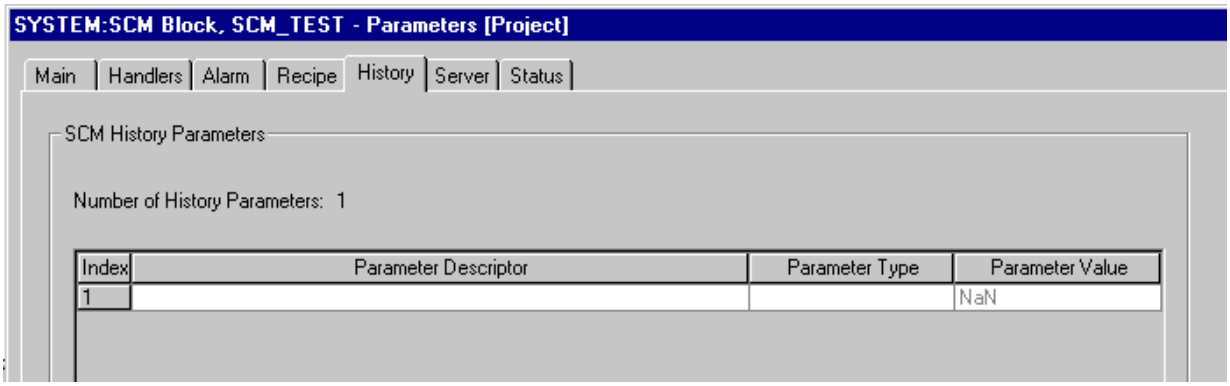
  Select whether you want to use a default range to automatically determine the low and high values for the specified target value or not. If not, you must enter low and high values individually.

- **Default Target Range**

  Specify range in percent to be used to automatically determine the low and high target values based on the entered target value.

History

Every SCM will allow the configuration of up to 50 different history values. Each parameter will include the following configuration data.



- **Descriptor**

  Description of up to 63 characters.

- **Type**

  Specify type of history parameter using 11 characters. You can enter this string or it can be determined by a batch application, if applicable. Some examples of history types are locations. (Where the ingredient is located), Inventory (how much was taken from stock), and Actual Value (some other recorded value for the process).

- **Parameter Value**

  SCM saves history parameter value as a real number.