
PlantScape Controller Implementation Book 2

**Release 400
11/01**

Release 400

Honeywell Confidential & Proprietary

This work contains valuable, confidential, and proprietary information. Disclosure, use, or reproduction outside of Honeywell, Inc. is prohibited except as authorized in writing. This unpublished work is protected by the laws of the United States and other countries.

Notices and Trademarks

**Copyright 1998 by Honeywell Inc.
Release 200 – November, 2001**

While this information is presented in good faith and believed to be accurate, Honeywell disclaims the implied warranties of merchantability and fitness for a particular purpose and makes no express warranties except as may be stated in its written agreement with and for its customers.

In no event is Honeywell liable to anyone for any indirect, special or consequential damages. The information and specifications in this document are subject to change without notice.

Honeywell, and **TotalPlant** are U.S. registered trademarks and **PlantScape** is a U.S. trademark of Honeywell Inc.

Other brand or product names are trademarks of their respective owners.

Honeywell Inc.
Industrial Automation and Control
Automation College
2820 West Kelton Lane
Phoenix, AZ 85023
(602) 313-4827

About This Document

References

The following list identifies all non-**PlantScape** documents that may be sources of reference for the material discussed in this publication.

Document Title	Doc. ID
<i>Knowledge Builder</i>	TC - DCDX11

Contacts

World Wide Web

The following lists Honeywell's World Wide Web sites that will be of interest to our industrial automation and control customers.

Honeywell Organization	WWW Address (URL)
Corporate	http://www.honeywell.com
Industrial Automation and Control	http://www.iac.honeywell.com
International	http://www.honeywell.com/Business/global.asp

Telephone

Contact us by telephone at the numbers listed below.

	Organization	Phone Number
United States and Canada	Honeywell Inc.	1-800-343-0228 Sales
	Industrial Automation and Control	1-800-525-7439 Service
	<i>Phoenix, Arizona U.S.A.</i>	
Asia Pacific	Honeywell Asia Pacific Inc. <i>Hong Kong</i>	(852) 8298298
Europe	Honeywell PACE <i>Brussels, Belgium</i>	[32-2] 728-2111
Latin America	Honeywell inc. <i>Sunrise, Florida U.S.A.</i>	(305) 364-2355

Unit 7

Introduction To Sequential Control Modules

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

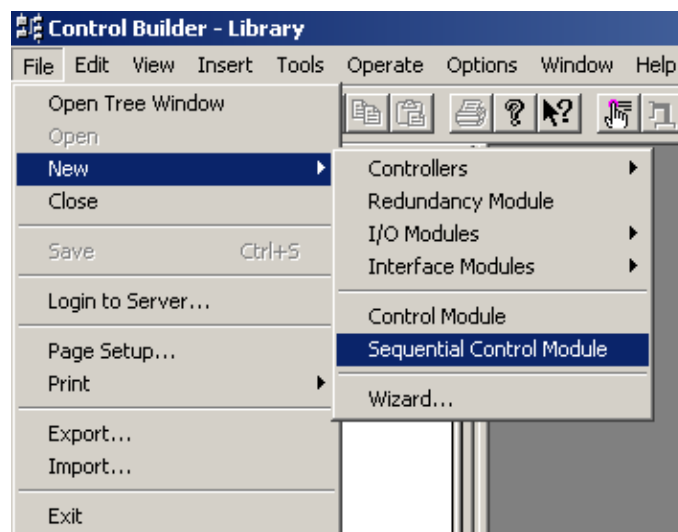
- Abnormal condition handling capability
- Computational capability
- SCM structure relates directly to the physical process
- Programming requirements are limited to modulating, sequential and logical control functions that relate directly to the process
- Statements, which are elements of a SCM, consist 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**

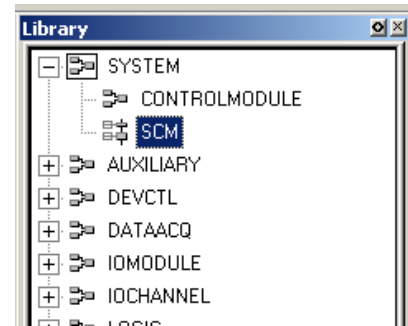


Creating a Sequential Control Module (SCM) ...continued

Second Method

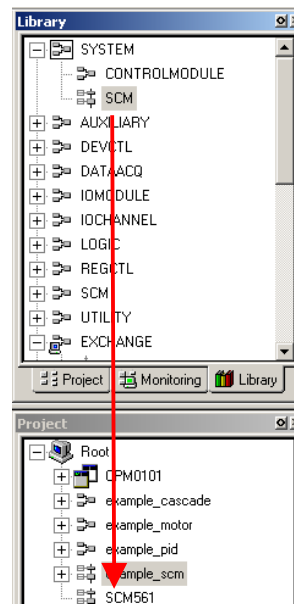
From the Library Tab expand the system directory. This will expose the option of creating a Control Module or an SCM,

Double Click on the SCM



Third Method

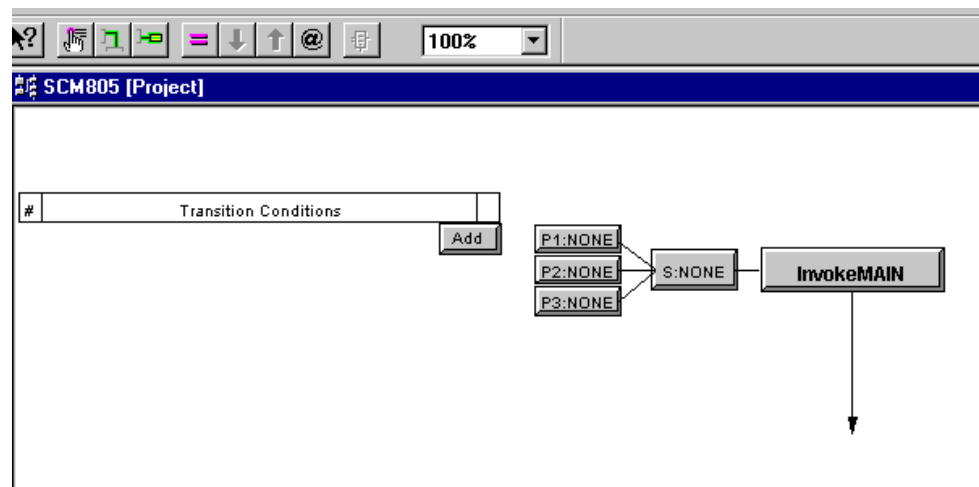
From the Library Tab expand the system directory. Open a second Tree View and select the project tab. Select SCM from the Library tab and drag it over to the Project tree.



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 SCM561).

Configuring an SCM's 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.

The screenshot shows the "SYSTEM:SCM Block, SCM805 - Parameters [Project]" dialog box. It has tabs for "Main", "Handlers", "Alarm", "Recipe", "History", "Server", and "Status". The "Main" tab is selected. The form contains the following fields:

Name:	<input type="text" value="SCM805"/>	Execution Period:	<input type="text" value="DEFAULT"/>
Description:	<input type="text"/>	Execution Phase:	<input type="text" value="-1"/>
Engr Units:	<input type="text"/>	Unit Text:	<input type="text"/>
Keyword:	<input type="text"/>	Version:	<input type="text"/>
Enable Alarming Option:	<input checked="" type="checkbox"/>		
Execution Order in CEE:	<input type="text" value="10"/>		

At the bottom, there is a checkbox labeled "Show Parameter Names" and three buttons: "OK", "Cancel", and "Help".

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

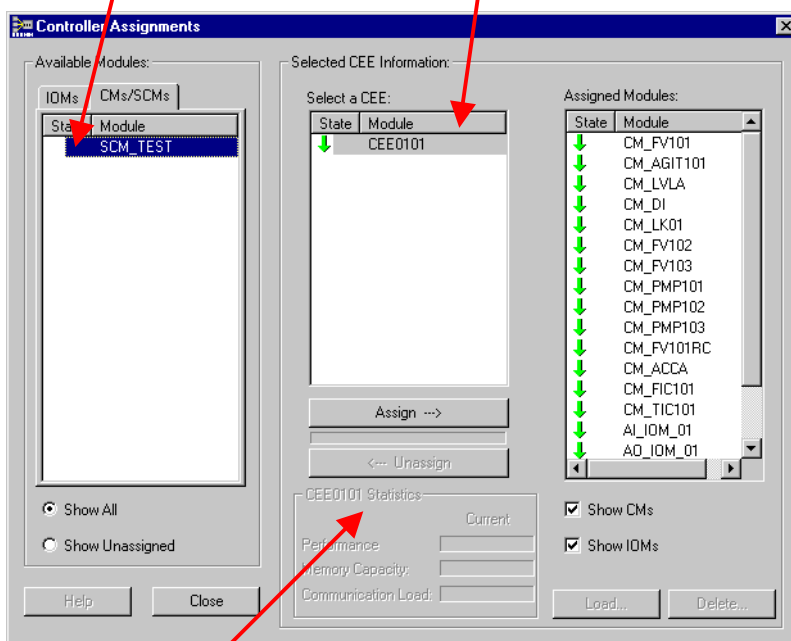
Before an SCM can be properly configured it must be assigned to a Control Execution Environment (CEE). A sequence has been established to facilitate the assignment process. Following this sequence should avoid problems during assignment.

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

<u>Types of handlers</u>	<u>Internal</u>	<u>Normal</u>	<u>Abnormal</u>
	(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

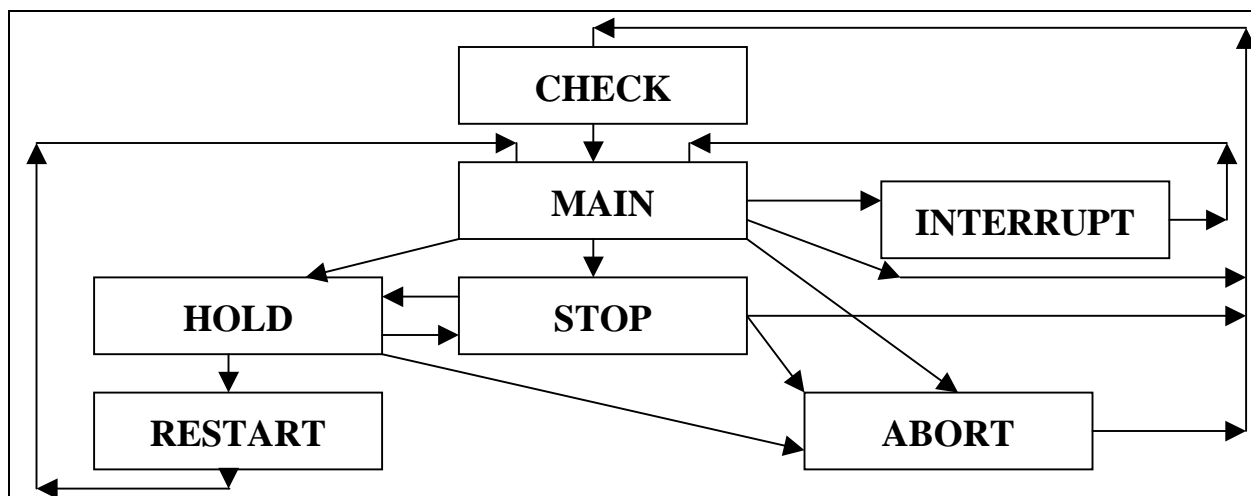
The **Stop Handler** preempts the activity of the Main/Interrupt, Restart or Hold Handlers. The Abort Handler can preempt the Stop Handler. From the Stop Handler you can go automatically to the Abort or Check Handler or by command to the Hold Handler.

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.



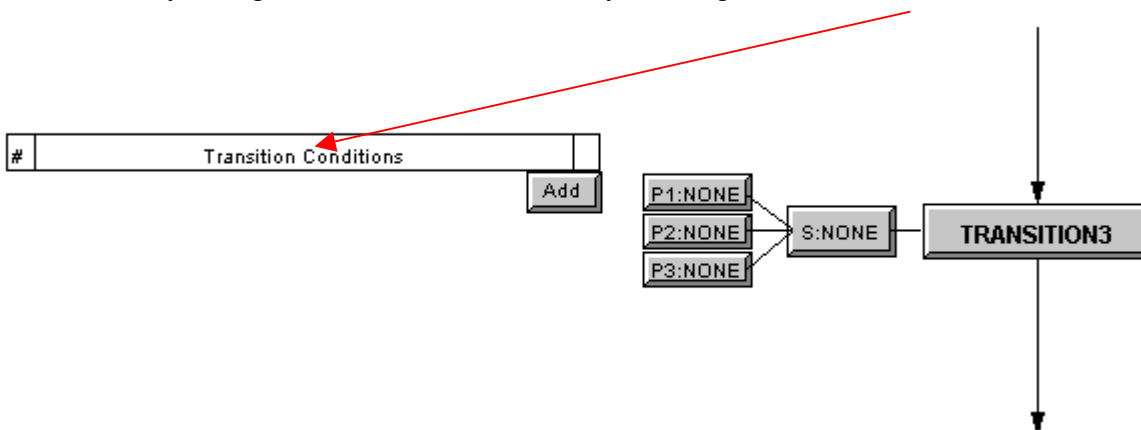
Creating a Transition

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.

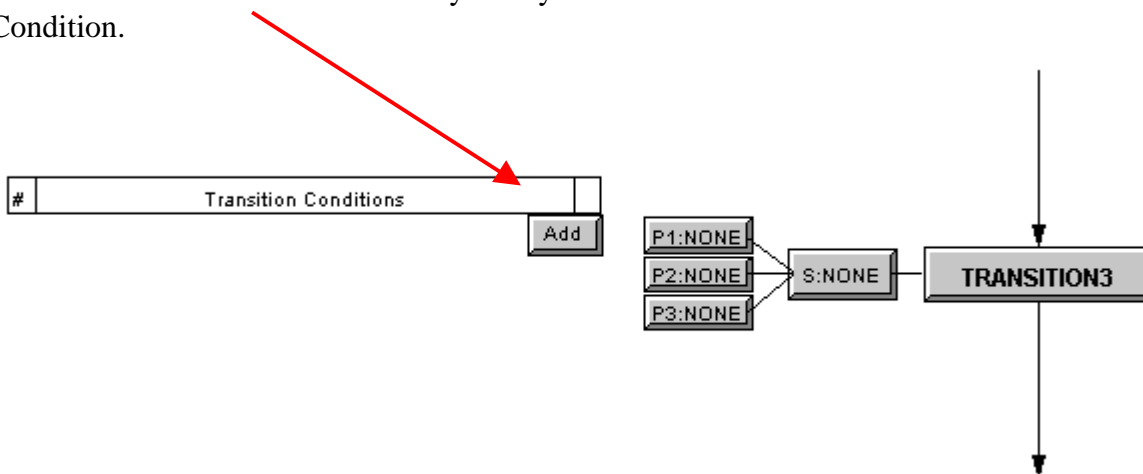


If you use the Default Invoke Transition as provided, the Handler requires an operator Start command each time that it is to run. If you want the Handler to start automatically due to process inputs, the Invoke Transition must contain statements that can be evaluated as true or false. If evaluated to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

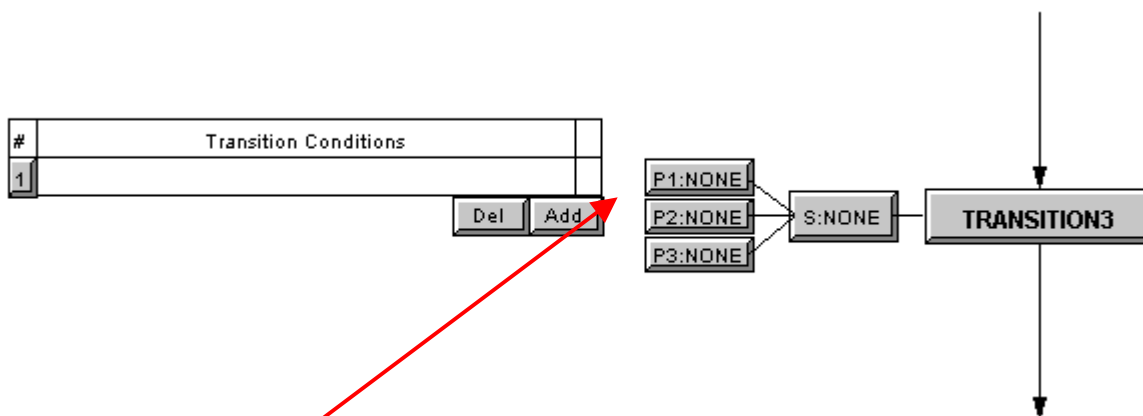
Creating a Transition ...continued

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

The screenshot shows the 'Main' tab of the 'SCM:TRANSITION Block, TRANSITION3 - Parameters [Project]' dialog. It has four sub-tabs: 'Main', 'Cond. #1', 'Gates', and 'Block Pins'. The 'Main' tab is active, showing a 'Name:' label with an empty text box and a 'Description:' label with a larger empty text box.

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

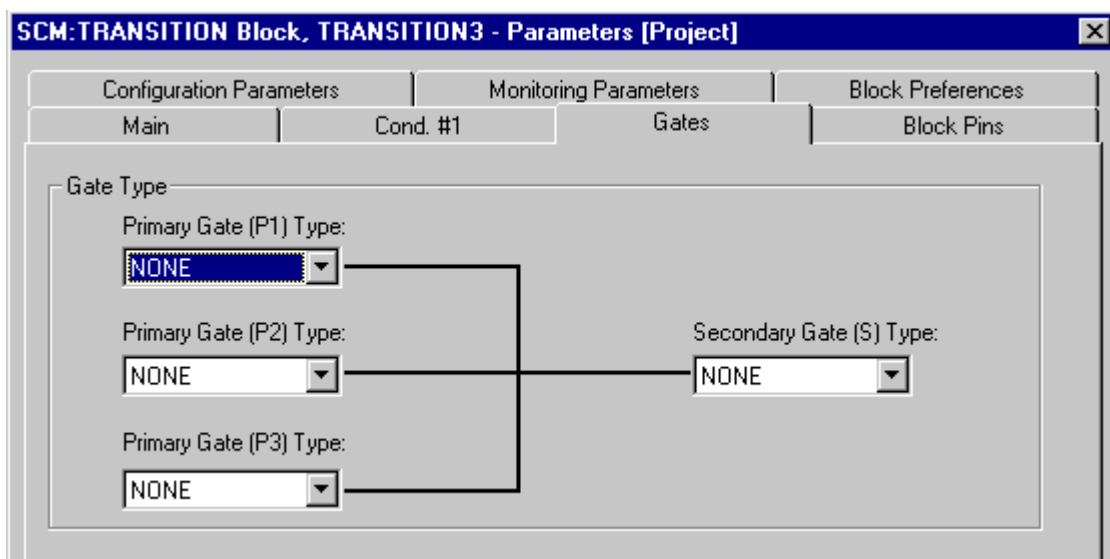
- Description field: Description of Condition you will be Implementing (23 characters maximum)
- Condition Expression field: Logical Expression to execute a function in Control Builder (written in "system" expression syntax)

The screenshot shows the 'Cond. #1' tab of the same dialog box. It features a 'Condition Details' section with a 'Description:' label and a text box. To the right of the description box are 'Force Permit' (checkbox) and 'Force Request:' (dropdown menu set to 'NONE'). Below the description is a 'Condition Expression (Ex: "A > B" or "C = n"):' label and a large text box. To the right of the expression box is a 'Logic Gate' section with three radio buttons: 'GATEP1' (selected), 'GATEP2', and 'GATEP3'. Below these are several buttons: '+', '-', '*', '/', '(', ')', '=', '>', '<', '>=', '<=', '<>', 'AND', 'OR', 'NOT', a numeric keypad (0-9, '.'), '^', '?', ':', 'Fetch Enums', 'Points...', 'No Enumerations:', and 'Functions:'. There are also two dropdown menus at the bottom right.

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)		

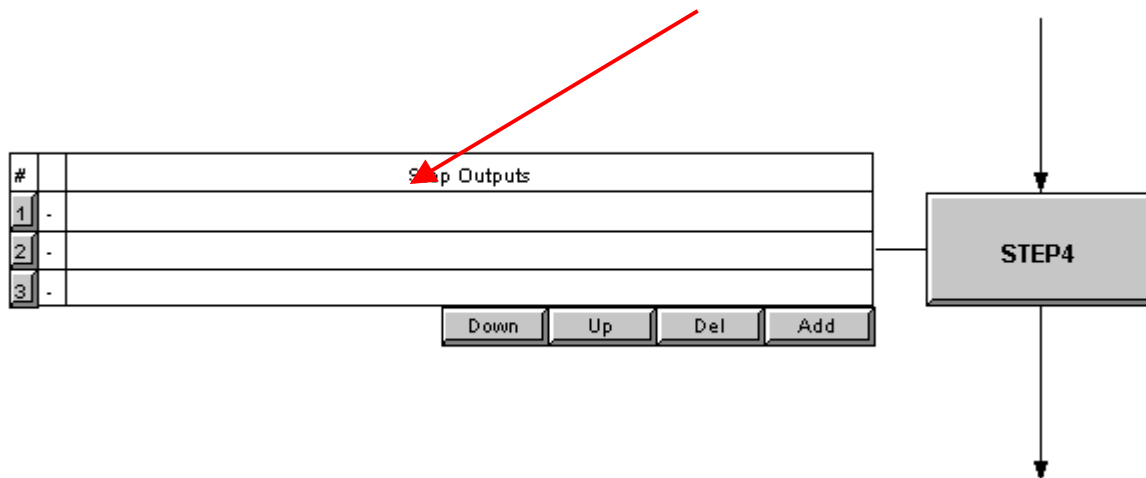
Creating a Step

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.

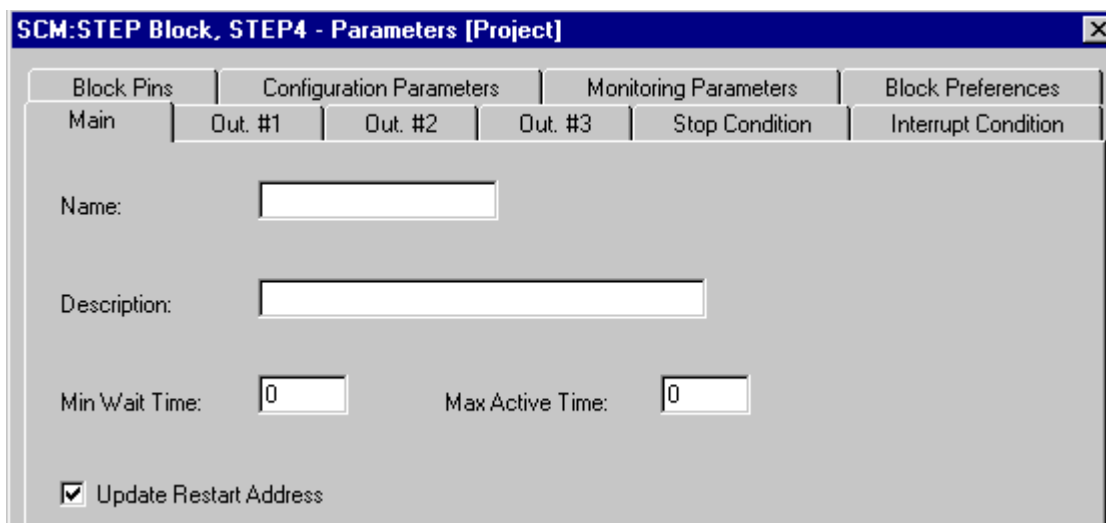


Creating a Step ...continued

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.

SCM:STEP Block, STEP4 - Parameters [Project]

Block Pins | Configuration Parameters | Monitoring Parameters | Block Preferences

Main | Out. #1 | Out. #2 | Out. #3 | Stop Condition | Interrupt Condition

Output Details

Description:

Type: NULL

Output Expression (Ex: "CM.Block.Param := A + B");

+ - * / [] = > < >= <= <> AND OR NOT

0 1 2 3 4 5 6 7 8 9 . ^ ? : :=

Fetch Enums No Enumerations: Functions:

Points...

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.

The screenshot shows the 'SCM:STEP Block, STEP4 - Parameters [Project]' dialog box with the 'Stop Condition' tab selected. The 'Condition Option' checkbox is unchecked. The 'Condition Details' section includes a 'Description' field, a 'Force Permit' checkbox, and a 'Force Request' dropdown menu set to 'NONE'. Below these is a large text area for the 'Condition Expression (Ex: "A > B" or "C = n")'. A numeric keypad and logical operators are provided for input. At the bottom, there are buttons for 'Fetch Enums', 'Points...', and dropdown menus for 'No Enumerations' and 'Functions'.

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.

This screenshot is identical to the one above, showing the 'SCM:STEP Block, STEP4 - Parameters [Project]' dialog box, but with the 'Interrupt Condition' tab selected. The layout and fields are the same, including the 'Condition Option' checkbox, 'Condition Details' section, and the numeric keypad.

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)		

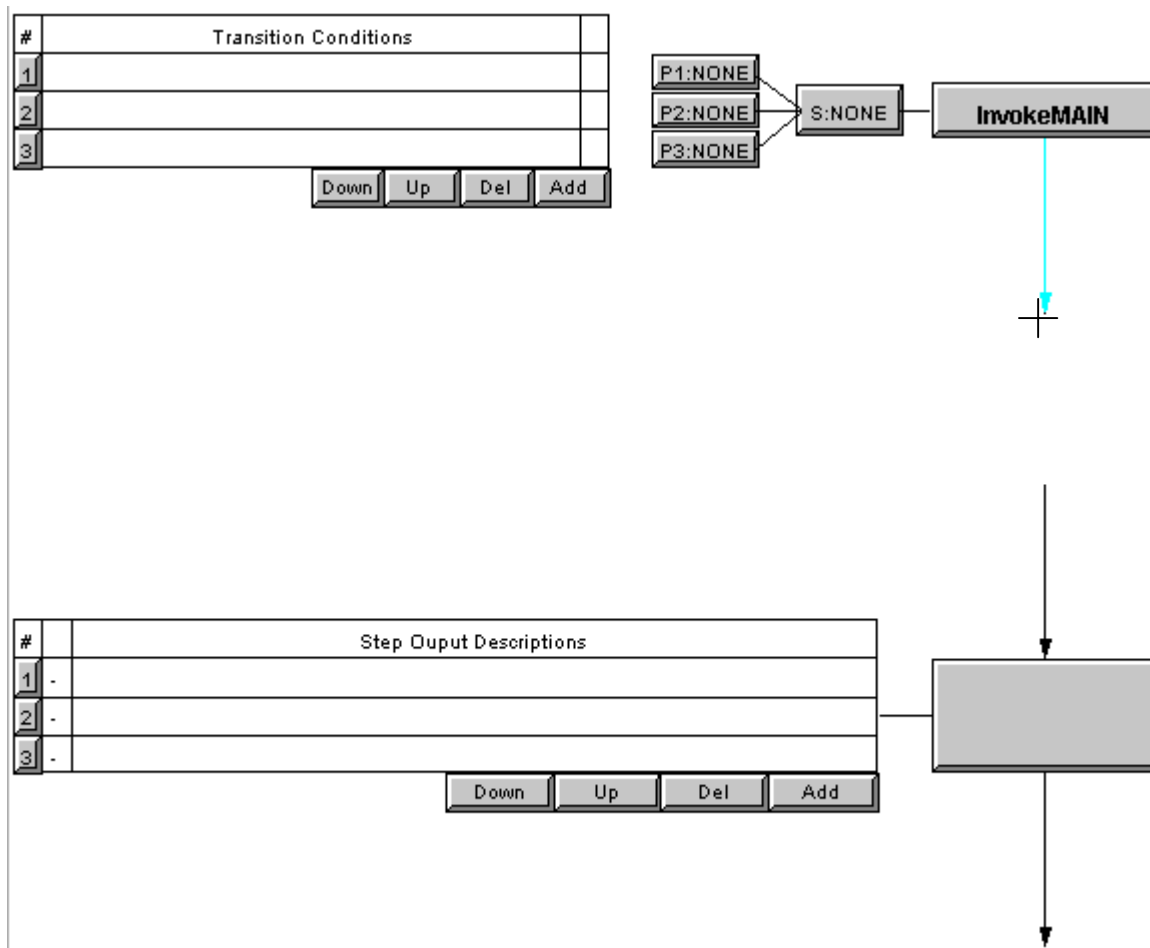
Wiring Within an SCM

Wiring an 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

SYSTEM:SCM Block, SCM_TEST - Parameters [Project]

Main | Handlers | Alarm | **Recipe** | History | Server | Status

SCM Recipe Parameters

Number of Recipe Parameters: 1 Default Target Range: +/- % ☐ Use Default Target Range

Index	Parameter Descriptor	Target Value	Target Hi	Target Lo	Matl Code	Scale
1		NaN	NaN	NaN	0	<input type="checkbox"/>

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.

The screenshot shows a software window titled "SYSTEM:SCM Block, SCM_TEST - Parameters [Project]". It has several tabs: Main, Handlers, Alarm, Recipe, History (selected), Server, and Status. The "History" tab displays "SCM History Parameters". Below this, it says "Number of History Parameters: 1". A table with four columns is shown: Index, Parameter Descriptor, Parameter Type, and Parameter Value. The first row has the index "1", an empty descriptor, an empty type, and the value "NaN".

Index	Parameter Descriptor	Parameter Type	Parameter Value
1			NaN

- **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.

Honeywell

PlantScape Controller Implementation

Lesson 2

Understanding the Temperature Control SCM

7 - 29

Notes

Introduction

The purpose of this Lesson is to give you knowledge of Sequential Control Modules. Upon completion of this Lesson you will have imported a Temperature Control SCM from a master database. You will have studied the code, loaded the SCM, and activated it.

Objectives

- ❶ Understand Control Builder Import\Export Functions
- ❷ Import a Temperature Control SCM
- ❸ Examine and understand the SCM code
- ❹ Load and Activate the SCM



Background for Export\Import Functions

- Your Project File can contain a great many items. It may contain IOMs, CMs, and SCMs for up to ten C200 controllers
- There is a tool which allows the copying of some or all of a project to insert into another project. That tool is the Export\Import tool.
- Export\Import is accessed from Control Builder
- Export copies the designated portion of the source project from Control Builder to text files
- Import copies from the designated text files into the destination Control Builder Project tab
- Both Import and Export have useful features which we will examine. We will then Import an SCM to our project.
- First we will examine the Export function

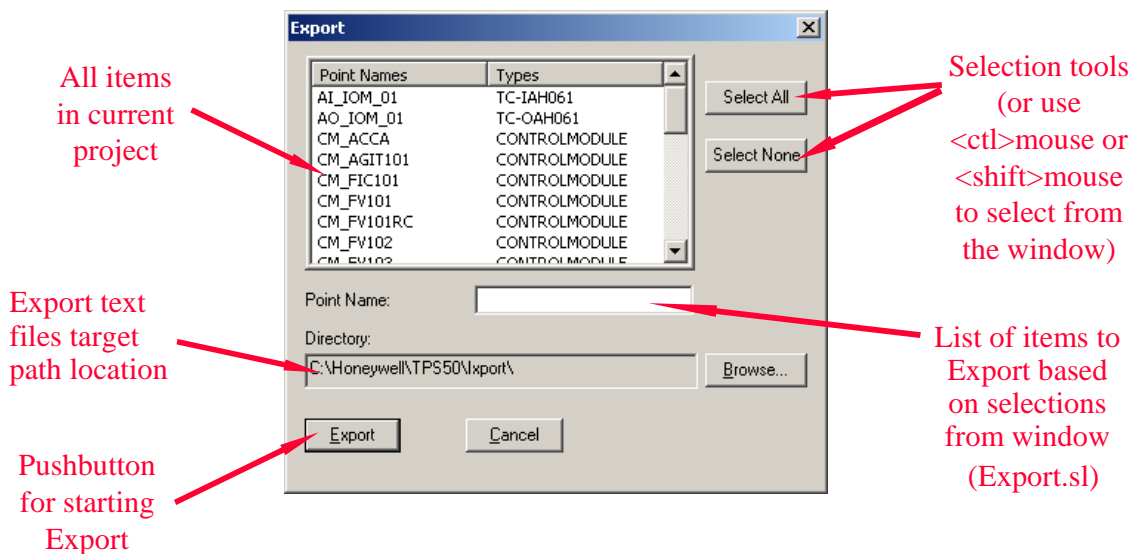
7 - 30

Notes

Honeywell

Export

- In Control Builder, select **File** → **Export** to call up the dialog box shown:



7 - 31

Notes

Export

The Export function copies designated Control Builder Project items to the directory shown in the Export dialog box. The items are converted into text files, one file for each function block along with other files used by the Import\Export function. Note: these files **cannot** be used for configuring similar objects using a text editor.

An additional file, Export.sl, is created. It is a selection list which contains the tag names of the items chosen to Export.

The default file location for the Export is Honeywell\tps50\Iexport

Honeywell

Export ...continued

To perform an Export:

- Scroll down in the Project window and find the object(s) you wish to Export to another Project file
- Select the object(s) to cause it to appear in the **Point Name** port
- Select the **Export** pushbutton to cause the object(s) to be put into text file format in the target path location.
- You can locate the resulting files in windows Explorer. There is a file for each Function Block as well as a file for the overall configuration. Also, there is a .bcd file used internally by PlantScape, and a Selection List called Export.sl. The text files are very small. (Your entire class project would fit on a single floppy in Export format.)
- We will now examine the Import Function

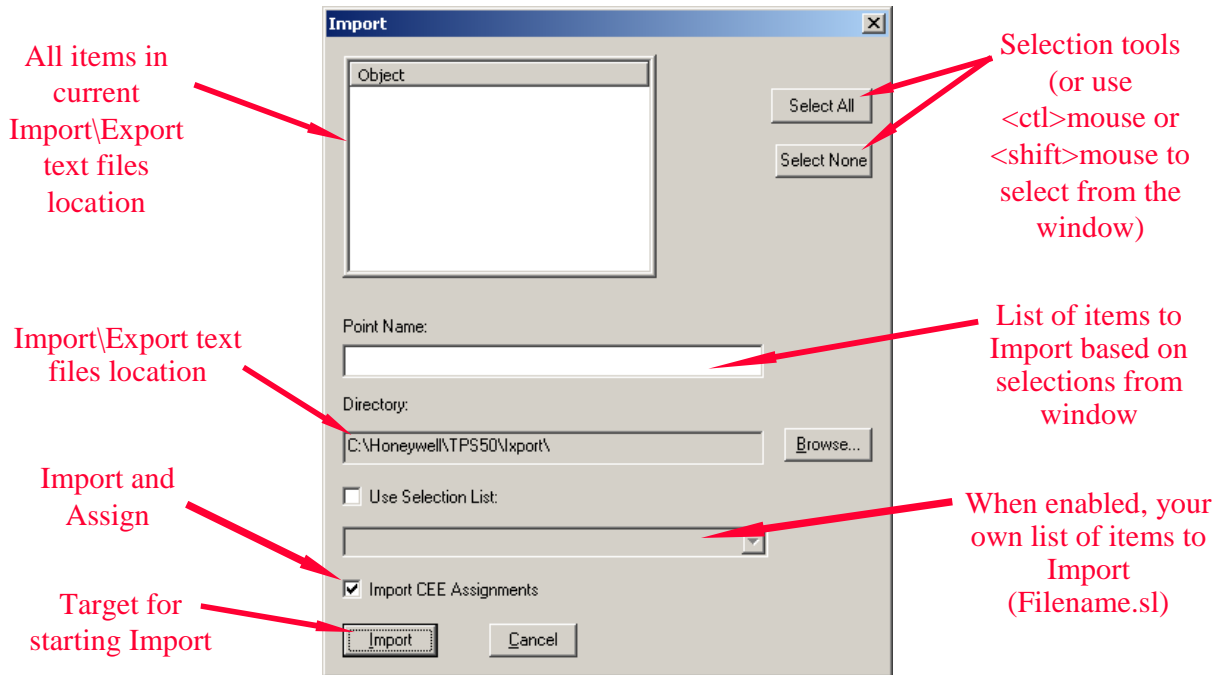
7 - 32

Notes

Honeywell

Import

In Control Builder, select **File** → **Import** to call up the dialog box shown:



7 - 33

Notes

Import

The Import function copies designated items from the Import\Export text files, located in the directory shown in the Import form, to the Control Builder Project.

The items to be Imported are listed in a file called a selection list. If items are selected from the Import form window, the list created is called Import.sl. You can use notepad to create your own selection list as well. To use this type of list, copy your .sl file to the same directory that contains the item text files, check the Use Selection List check box, and browse to the list file. The items contained in the list will appear in the Point Names port.

If the items were assigned in the source project file, and the same controller name exists in the target project file, assignment to the controller can be done at Import. If this is not desired, do not check the Import CEE Assignments check box. The Imported Items will then appear under the project root.



Import ...continued

To perform an Import:

- Select the **Browse...** pushbutton and find the object(s) you wish to Import to your Project
- Select the object(s) to cause it to appear in the **Point Name** port
- Uncheck the **Import CEE Assignment** check box if you want to assign the object after it is added to Project
- Select the **Import** target
- The object will now be in your project under the Root, since Assignment was not Imported
- The object can now be modified, assigned, and loaded to a controller
- We will now Import an SCM to our Project root

7 - 34

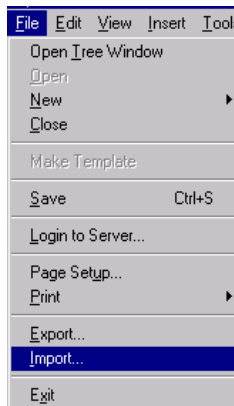
Notes

Honeywell

Importing Sequential Control Modules

Open the Import Tool

- Open Control Builder, if not already open
- Click **File** → **Import** to call up the Import Dialog box shown on the next page



7 - 35

Notes

Overview

Sequential Control Modules (SCMs) are made up of **Handlers**. Handlers are made up of **Transitions** and **Steps** that execute a procedural process. Steps direct CMs to take actions after Transition conditions are met. The actions of one Step cannot occur until a previous Step has been completed and the previous Transition conditions are met. SCMs execute their processes by monitoring and controlling previously configured CMs.

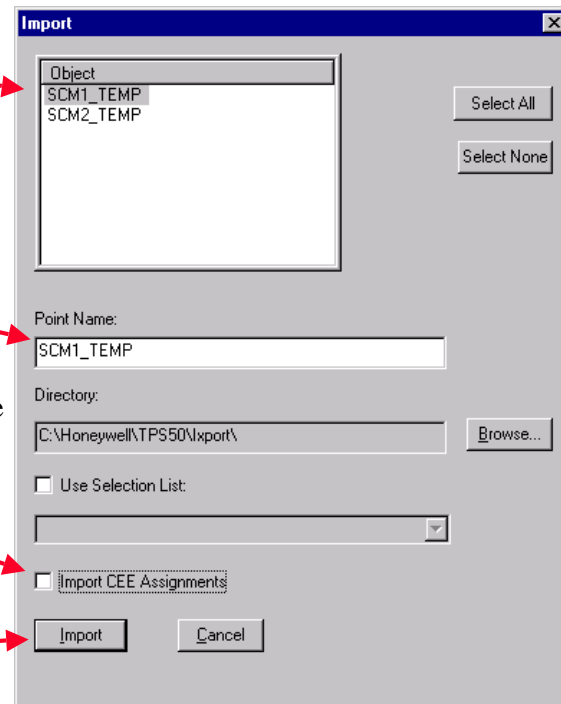


Refer to the *Control Builder Components Theory* for a detailed explanation of the differences and interfaces between Control Modules and Sequential Control Modules.



Importing a Sequential Control Module ... *continued*

- In the Object window, select the appropriate SCM to Import*
Team1: **SCM1_TEMP**
Team2: **SCM2_TEMP**
- Note that your choice appears in the Point Name Window
- Uncheck Import CEE Assignments so that the SCM will be imported under the Project root
- Click the Import button to add the SCM to your Project



7 - 36

Notes

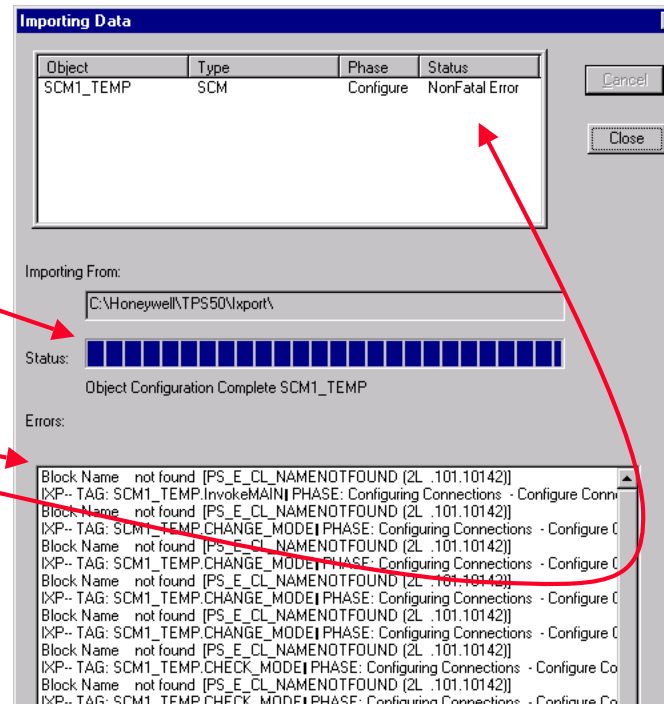
Importing SCMs

*The reason that the two SCMs appear in the Object window is because they were Exported from a master erdb to your **C:\Honeywell\TPS50\Ixpport** directory prior to class.



Importing a Sequential Control Module ... continued

- As the Import progresses, status is shown in the Importing Data dialog box which appears after the Import button is selected.
 - The Status bar indicates Import function progress
 - If errors are encountered, they display as they occur in the Errors window
 - At Import completion, the Status of the Import is shown in the top window. In this example, there were many errors, but they were NonFatal meaning the Import took place. **You should get no errors.**
- Click Close



7 - 37

Notes

Importing SCMs -- Errors

Import errors can be numerous when importing SCMs. If the CMs referenced by the SCM being imported do not already exist in Project, every reference to a non-existent CM will create an Import error. That is what happened in the above example -- referenced CMs had not yet been added to Project.

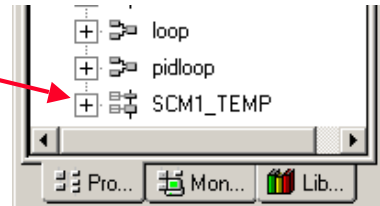
These errors are nonfatal, meaning the Import still takes place, but the errors must be corrected prior to operation or the SCM will fail.

You should have all referenced CMs in Project at this time and you should get no errors. **See your Course manager if you do get errors.**



Examining a Sequential Control Module

- The SCM should now appear in the Project under the root.
- Open the SCM and study the code. Compare it to the process description in the notes below. Click the top line of each Transition and Step to alternate between Description and Code Syntax.
- Double click on Steps and Transitions to view configuration detail. Note the Primary and Secondary Gate configuration in Transitions.
- See your Course manager for any clarification you may need. When you understand the code of the SCM go to the next page. We will assign, load, and activate the SCM.



7 - 38

Notes

Temperature Process Description

This SCM will operate the CM#_TIC101 CM to heat and cool the Reactor as follows:

1. Change CM#_TIC101 & CM#_FIC101 Mode Attributes to Program, Modes to Auto and Cascade.
2. Verify Mode settings
3. Change CM#_TIC101 SP to 80.0 Degrees.
4. Wait until CM#_TIC101 PV is within 2 Degrees of SP.
5. Change CM#_TIC101 SP to 35.0 Degrees
6. Change CM#_TIC101 Mode Attribute to Operator
7. Wait until CM#_TIC101 PV is within 3 Degrees of SP.

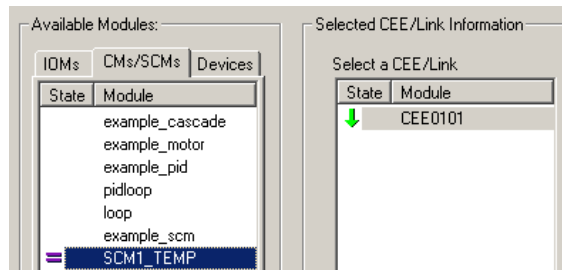
CM#_FLAGS.TEMP is the automatic start for this SCM.



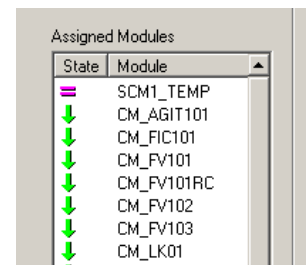
Assigning SCMs to the CEE

- From the Control Builder **Tools** menu, click **Assign** to display the Controller Assignments

- Locate the Available Modules section on the left side of the window and click **SCM #_TEMP** in the tab labeled **CMs/SCMs**



- Select the destination CEE (CEE0101)
- Click **Assign** in the center of the window and note that your SCM appears in the **Assigned Modules** window after a few seconds
- Click **Close**



7 - 39

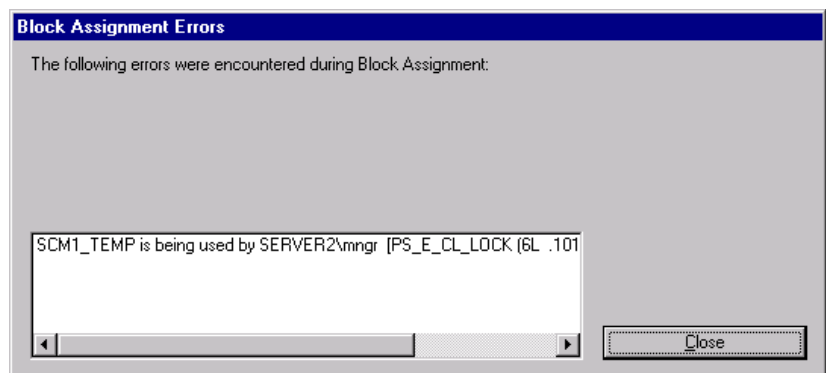
Notes

Assigning SCMs to the CEE

Before you can load your SCM it must be assigned to a CEE.

Please note the following:

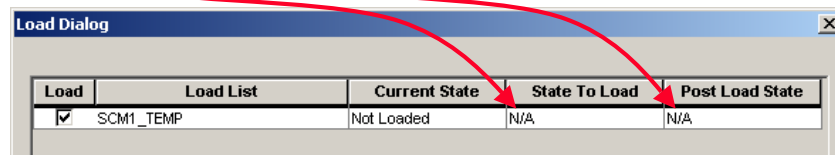
- If you get the Block Assignment Error shown, close the SCM in the edit window and try again. A CM, IOM or SCM cannot be assigned if it is open for editing.



Honeywell

Loading and Activating the SCM

- Open the **Project** Tab in Control Builder
- Select the **SCM#_TEMP**
- Right click and select **Load...**
- Verify that the SCM appears correctly in the **Load Dialog** box. Note that unlike IOMs and CMs, SCMs cannot be inactivated and reactivated automatically by the load function.
- Click **OK**



- In the Monitoring tab, right click on the **SCM** and select **Activate**
- In **Station**, add **SCM#_TEMP** to **Group #2, Slot 5**

7 - 40

Notes

Loading the SCM

Selecting **Load...** causes the **Load Dialog** box to appear, which lists the objects to be loaded. The list can be modified as only those objects checked in the **Load** column will be acted upon. When you select **OK** the load begins. If there are any errors, a message box will appear to give you details of the problem(s).

Once the load is complete, all of the loaded objects appear in the Monitoring Tab. They are loaded to your controller and to the PlantScape Server.

SCMs, like all objects in the Monitoring tab can be many colors. Each color represents a different status. When an object is:

- **green**, the object is running or active
- **blue**, the object is not running, or inactive
- **red**, the object in the Monitoring Database differs from its counterpart in the C200.

Honeywell

This completes....

PlantScape Controller Implementation

Lesson 2

**Understanding the Temperature Control
SCM**

7 - 41

Notes

Honeywell

PlantScape Controller Implementation

Lesson 3

Operating Sequential Control Strategies

7 - 43

Notes

Introduction

The purpose of this Lesson is to give you the knowledge to be able to operate your newly loaded SCM in Control Builder and Station. Upon completion of this Lesson you will have operated the SCM#_TEMP.

Objectives

- ❶ Start, Stop, and Reset SCM#_TEMP in Control Builder
- ❷ Start, Stop, and Reset SCM#_TEMP in Station
- ❸ Monitor status of SCM#_TEMP in Control Builder and Station



SCM states

- Control Builder uses State descriptors to indicate the status of a Sequential Control Module

State	Description
IDLE	SCM is ready to begin execution of its Main Handler
INACTIVE	SCM Block EXECSTATE is INACTIVE; block configuration parameters can be edited and downloaded
VALIDATED	SCM Block EXECSTATE is Active
RUNNING	SCM is executing its Main Handler
INTERRUPTING	SCM is executing its Interrupt Handler.
ABORTED	SCM has executed its Abort Handler
HELD	SCM has executed its Hold Handler
STOPPED	SCM has executed its Stop Handler
COMPLETE	SCM has finished its Main Handler

7 - 44

Notes

Connecting and disconnecting SCMs



For a full description of all the SCM States and how an SCM transitions from one to another, see *Control Builder Components Theory, Sequential Control*.

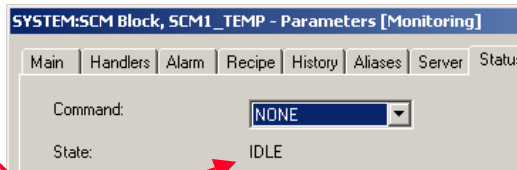
Honeywell

Operate Your SCM

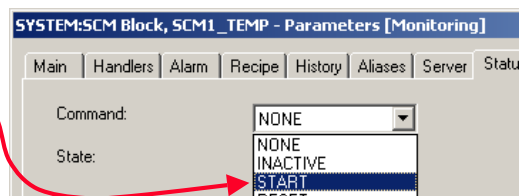
- Select **SCM#_TEMP** from the **Monitoring** tab. Right click, then select **Configure Module Parameters**.

- Click the **Status** tab

- Notice the **State** is **IDLE**



- Click on the **Command** pull down menu and select **START**



- Close the Module Parameters dialogue box

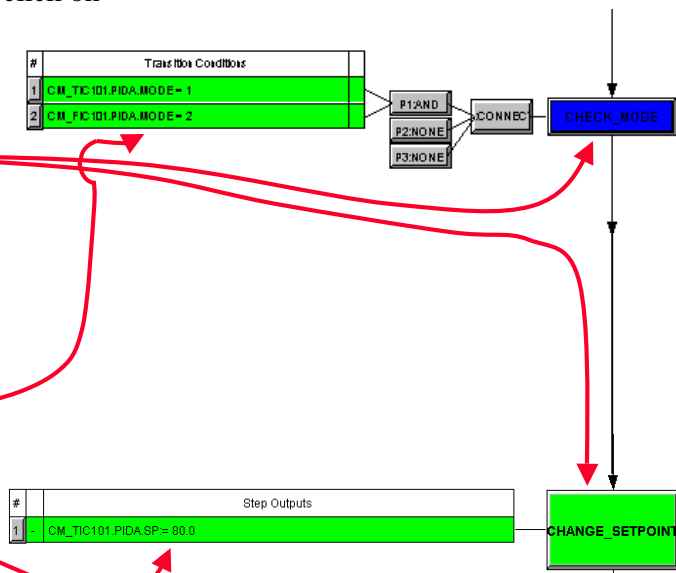
7 - 45

Notes

Honeywell

Operate Your SCM

- Back in the **Monitoring** tab, double click on **SCM#_TEMP**
- Notice the colors in the SCM
 - **Function Blocks:**
 - **Blue** Completed
 - **Green** Active
 - **Grey** Not Active yet
 - **Transition Statements:**
 - **Green** Condition is true
 - **Step Statements:**
 - **Green** Action complete
- Status should go to **Complete** after the SCM is finished
- To run again, issue a **RESET** (State to IDLE) then issue the **START** command



7 - 46

Notes

Honeywell

Opening an SCM Detail Display from Station

- In your Station's Command Zone, type the name of your SCM, **SCM#_TEMP** and press <F12> to call up the Detail of your SCM
- Operate your SCM using the **Command** pull-down:
 - **RESET**
 - **START**
- Click on the **Chart** tab to monitor run status

Note: You can also operate **SCM#_TEMP** from **Group #2**

SCM1_TEMP
TEMPERATURE CTRL SCM

SCM1_TEMP.MAIN

Active Handler
MAIN
Handler Type

Active Step
0
Time in Step (cyc)

ST COMPLETE
EXEC STS OK
CMD NONE
MD AUTO
MD ATTR OPERATOR

7 - 47

Notes

SCM Detail Display

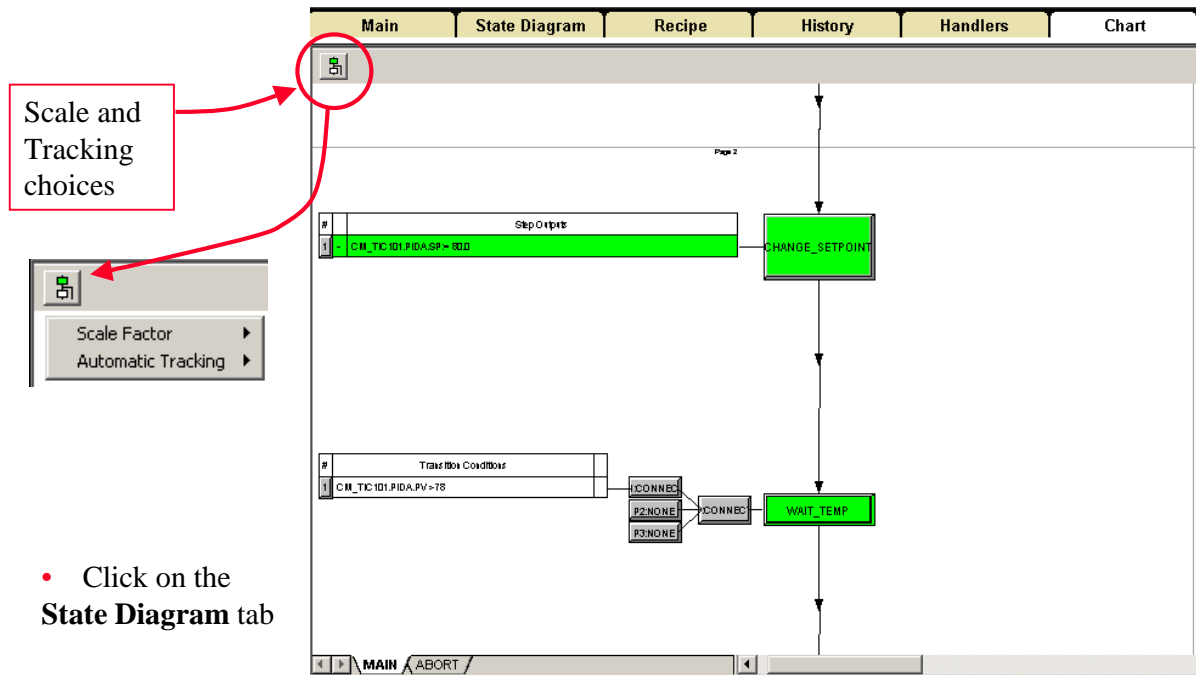
Operating from Control Builder can be useful, especially for troubleshooting and debugging, but for day-to-day operations you will probably prefer to use PlantScape's Station software.

The the Detail display contains several pages of SCM parameters including currently Active Handlers, Recipe data, and History data. Recipe values and Active Handler choices can be modified from station.



SCM Detail Display

- The **Chart** page is an active window into the **Monitoring** view of **Control Builder**. It is very useful in status monitoring and trouble-shooting since it has all the functionality of the Control Builder Monitoring window.



- Click on the **State Diagram** tab

Notes

SCM Detail Display -- Chart Page

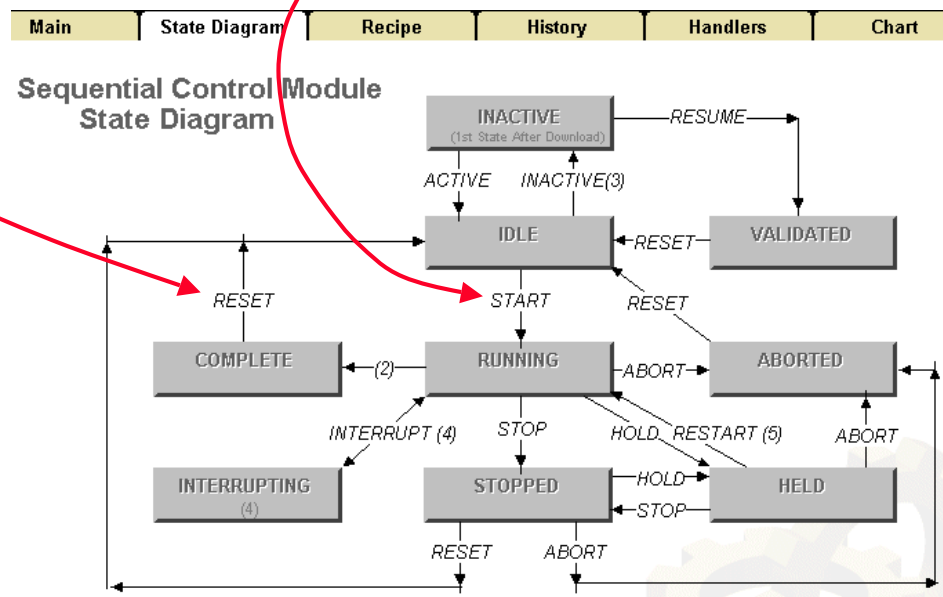
The Chart page of the Detail is a window into the Control Builder Monitoring view of the SCM. It is setup to track the active Step and Transition. The tracking feature can be turned off if required. This page is very handy for status checking and basic troubleshooting.

If the detail is called up while logged into Station as Supervisor or higher, the Chart page can be used to command the SCM, the same way you would from Control Builder.

Honeywell

SCM Detail Display

- The **State Diagram** page shows the Command\State flow chart. All actions involving SCMs must follow this diagram. Recall that to rerun **SCM#_TEMP** from **COMPLETE**, we had to **RESET** to go to **IDLE**, and then **START** to go to **RUNNING**



7 - 49

Notes

SCM Detail Display

Page two of the SCM Detail Display is the State page. It shows a very useful State/Command diagram. It is designed to guide you through the operation of the SCM.

If you are issuing **INACTIVE** commands to the SCM and nothing is happening, go to this page and verify that the SCM is in the proper state to accept the **INACTIVE** command. If the SCM is not in **IDLE**, it cannot be set **INACTIVE**. The chart shows how to get to the **IDLE** state.

Honeywell

This completes....

PlantScape Controller Implementation

Lesson 3

**Operating
Sequential Control Strategies**

7 - 50

Notes

Unit 7 Exam

QuesNo	Question
1	<p>What are the three Function Block types used in configuring SCMs</p> <p>A AUXCALC</p> <p>B STEP</p> <p>C FLAG</p> <p>D TRANSITION</p> <p>E HANDLER</p>
2	<p>What is the maximum number of outputs a Step can have?</p> <p>A 8</p> <p>B 10</p> <p>C 12</p> <p>D 16</p>
3	<p>What is the maximum number of conditions a Transition can have?</p> <p>A 8</p> <p>B 10</p> <p>C 12</p> <p>D 16</p>

- 4 When a new SCM is created an Invoke Transition is always added for you. Which of the following is true?
- A Every SCM must start with a Transition named Invoke Main.
 - B Every SCM must start with a Transition, the name is configurable.
 - C The Invoke Main Transition is added to keep Control Builder from fragmenting the PC hard drive.
 - D The Invoke Transition allows automatic startup of the SCM.
 - E Invoke Transition Conditions are only active when the SCM is in the IDLE state
- 5 When operating an SCM you must issue a RESET command to go from the COMPLETE state to the IDLE state.
- A True
 - B False
- 6 SCMs may be configured and activated in Control Builder but can only be operated from Station?
- A True
 - B False

7 SCMs can only be Started from the IDLE state.

A True

B False

Unit 8

SCM Functionality

Honeywell

PlantScape Controller Implementation

Lesson 1

Creating an SCM (SCM#_XFERA)

8 - 3

Notes

Introduction

The purpose of this Lesson is to give you the knowledge to be able to configure Sequential Control Modules. Upon completion of this Lesson you will have Created, Activated and Operated the Tank A Transfer SCM.

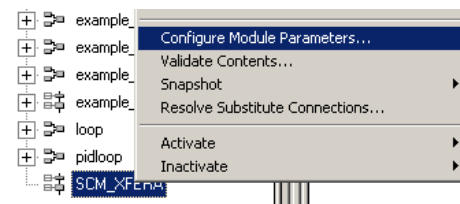
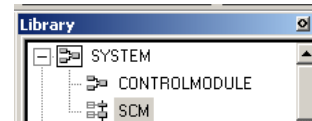
Objectives

- ❶ Create a New SCM
- ❷ Add Steps and Transitions
- ❸ Wire Steps and Transitions
- ❹ Load, Activate and Operate the SCM



Create a New SCM

- Set up Control Builder with both the **Library** and **Project** views visible
- Click and expand the SYSTEM Library under the **Library** Tree View
- Drag a new SCM from the **Library** to the **Project** view
- Name the SCM **SCM#_XFERA**
- Right-click on the SCM, and click on **Configure Module Parameters**
- Enter the following information on the **Main** tab
 - Description **TANK A TRANSFER**
 - Key Word **A TRANSFER**
 - Execution Period **Default** (1000 ms)
 - Execution Phase **-1** (defaults to 0)
- Close, Save, and Assign the SCM to **CEE0101**



8 - 4

Notes

Tank A Transfer Description

The Tank A transfer SCM automates the transfer of Ingredient A to the Reactor.

The process is as follows

- The Tank A totalizer **CM#_ACCA.TOTAL_A** is stopped and reset, and a charge amount is entered
- The totalizer is started and the Tank A bottom valve (**CM#_FV101**) is opened
- The A transfer regulatory control valve (**CM#_FV101RC**) is opened to 50% and the A pump (**CM#_PMP101**) is started
- The transfer flow is regulated to the required SP and continues until the target amount is reached
- The pump is stopped, the regulatory control valve is closed, and the bottom valve is closed



Start Conditions

- From the **Project** Tree View, double-click the **SCM** to open it
- Configure the following information on the **Invoke Transition** (If needed, see the next pages for configuration keystrokes and mechanics.) In the tables that follow, only add the Statements shown. This Transition has only one Condition Statement using gate P1.

Tab	Name	Description
Main	INVOKE_MAIN	INVOKE MAIN
	Description	Condition Expression Gate
Condition #1	XFERA FLAG ON	CM#_FLAGS.XFER_A.PVFL P1
Condition #2		
Condition #3		
Condition #4		
Gates	Pri Gate (1)	Pri Gate (2) Pri Gate (3) Secondary Gate
	CONNECT	CONNECT

- Click **OK**; Skip to page 8-15 if you already know how to Configure Transitions and Steps. The next several pages go through the mechanics of Step and Transition configuration

8 - 5

Notes

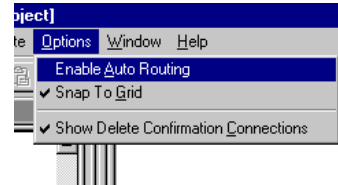
Start Conditions

There is one start condition for XFER_A. It looks at the **XFER_A** flag block in **CM#_FLAGS**. When the flag is on, the SCM starts.

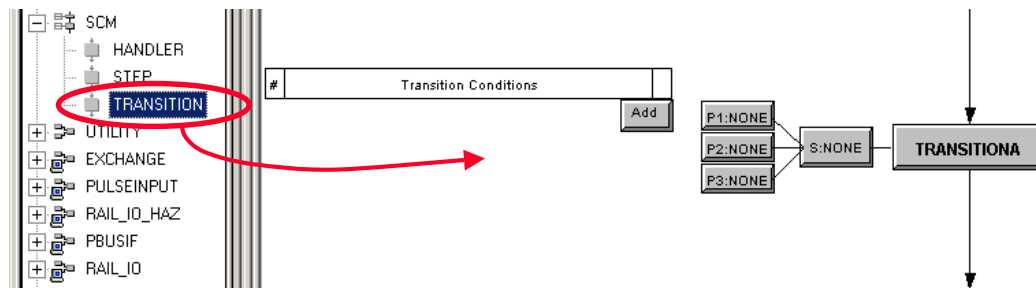
Honeywell

Creating Transitions

- On the Control Builder menu, click
 - Options**
 - Snap to Grid**Then
 - Options**
 - Enable Auto Routing** to facilitate building the SCM



- To configure a **Transition**:
- Drag a **Transition** function block to your SCM drawing area



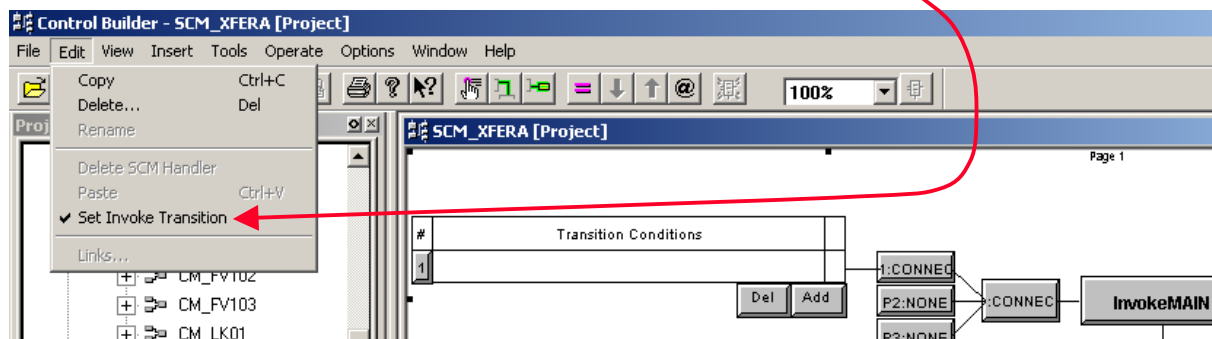
8 - 6

Notes

Transition blocks

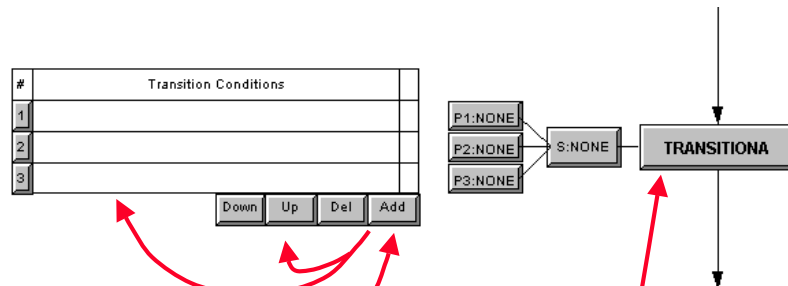
When a new SCM is added to a Project, the Invoke Transition is added automatically. All other Transitions must be added from the library.

To verify that a Transition is the Invoke Transition for a handler, single click the Transition block to highlight it. Then click **Edit** to see the **Set Invoke** check mark.



Honeywell

Creating Transitions



- Click the **Add** target once for each Condition Statement you wish to add. The Condition Statement ports appear along with the **Del**, move **Up**, and move **Down** targets
- Double-click on the **Transition** block to open the Configuration Dialog boxes

8 - 7

Notes

Condition Statements

The move Up and Down targets allow condition statements to be moved relative to each other. This is important because the conditions are evaluated from top to bottom in order. In Steps, the outputs are executed from top to bottom in order.



Creating Transitions

- In the Main tab, enter the Transition **Name** (16 characters max., no spaces) and **Description** (40 Characters max., spaces allowed; 19 visible in an SCM printout)

Main	Cond. #1	Cond. #2	Cond. #3	Gates
Name: CHECK_AUTO				
Description: CHECK MODE AUTO				
Configuration Status: OK				

- Next configure the Condition Statements by clicking the **Cond. #n** tab for each added condition

8 - 8

Notes

Name and Description

The **Name** of a Transition or Step is part of its code and must contain no spaces. The **Description** on the other hand is not code and is included only for clarification.

Honeywell

Creating Transitions

- The **Condition Details** configuration dialog box contains tools to allow configuration from menus. The general layout is shown below. Details are covered on the next pages.

- The **Description** requires keyboard entry (64 characters max., 43 visible in printout)
- The **Condition Expression** is the actual code of the statement and can be typed or configured using the tools:
 - Keyboard-type entry buttons
 - Enumeration listings
 - Point selector

Condition Details

Description:

Condition Expression (Ex: "A > B" or "C = n"):

Force Permit ☐ Force Request: NONE

Logic Gate: GATEP1, GATEP2, GATEP3

Configuration Status: OK

Configuration Description:

Execution Status: OK

Execution Description:

Keyboard interface buttons: +, -, *, /, [], =, >, <, >=, <=, <>, AND, OR, NOT, 0-9, ^, ? : , Fetch Enums, Points..., No Enumerations, Functions.

8 - 9

Notes

Configuration Tools

It is possible to add Condition and Output expression code without using the keyboard. The point database is available for point/parameter selection whether you are working at the Server or working remotely.

The size of the expression is decided by the configuration engineer. By using the pre-configured functions such as square root, the math operators $+$, $-$, $*$, $/$, the logical operators AND, OR, NOT, and the IF-THEN-ELSE functionality $? :$, the expression can contain a large number of characters. However, only the first 68 will be visible on a printout.

Honeywell

Creating Transitions

- We will now configure a sample Condition statement to demonstrate the use of the tools.
 - For our Condition we will verify that **CM_FIC101** is in **AUTO** mode.
- First we will use the Point selector (This tool is the same one we used for CM parameter connectors.)

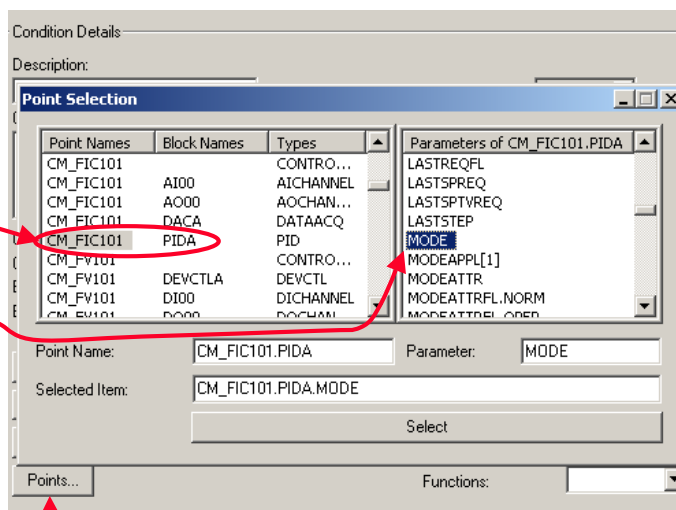
– Click the **Points** target to call up the **Point Selection** dialog box

– Scroll to **CM_FIC101.PIDA**; click it

– Scroll to **MODE**; click it

– Click the **Select** target

– Close the **Point Selection** dialog box



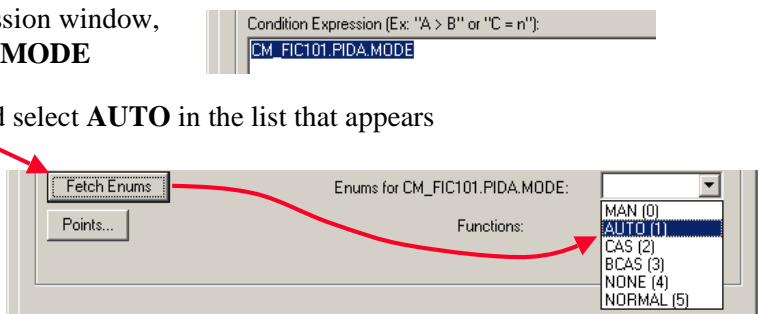
8 - 10

Notes

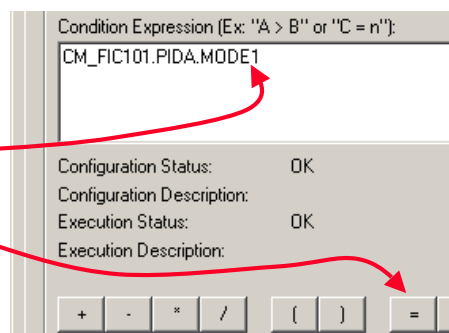
Honeywell

Creating Transitions

- Next we will use the Enumeration listing tool
 - In the Condition Expression window, highlight **CM_FIC101.MODE**
 - Click **Fetch Enums** and select **AUTO** in the list that appears



- Note that a 1 now appears in the Expression window
- Click the cursor between **.mode** and the **1** and click the **=** target



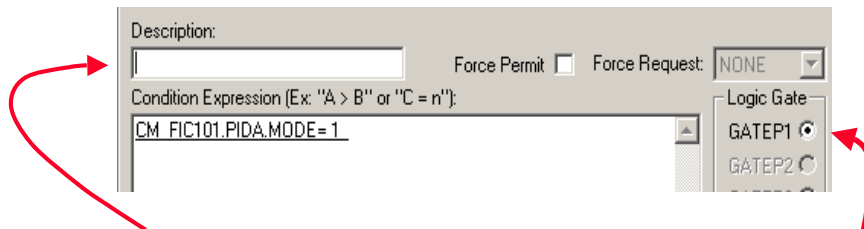
8 - 11

Notes

Honeywell

Creating Transitions

- The expression is now complete
- Click the <Tab> key to have Control Builder verify syntax.
An underline indicates correctness.



- Enter a condition **Description** up to 23 characters (You can do this prior to the Expression if desired.)
- Assign Primary **GateP1**
- The only thing remaining is to configure the **Primary** and **Secondary Gates**
 - Click the Gates tab to call up the configuration dialog box.

8 - 12

Notes

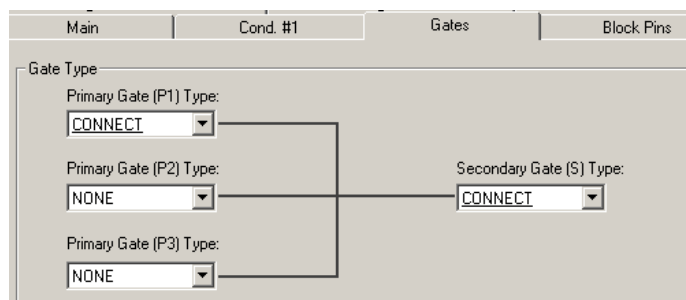
Expression and Description

The **Condition** or **Step Expression** is code and must follow syntax rules. The **Description** on the other hand is not code and is included only for clarification.



Creating Transitions

- A Transition can have a maximum of ten Conditions, assigned to up to three Primary gates, which all feed into one Secondary gate.
 - In our example, we have only one Condition assigned to Primary gate **P1**
 - With only one condition, choose gate type **Connect**
 - Since we have only one Primary gate, the Secondary will also be of type **Connect**
- Leave the Gates P2 and P3 defaulted to **None** and click **OK**



8 - 13

Notes

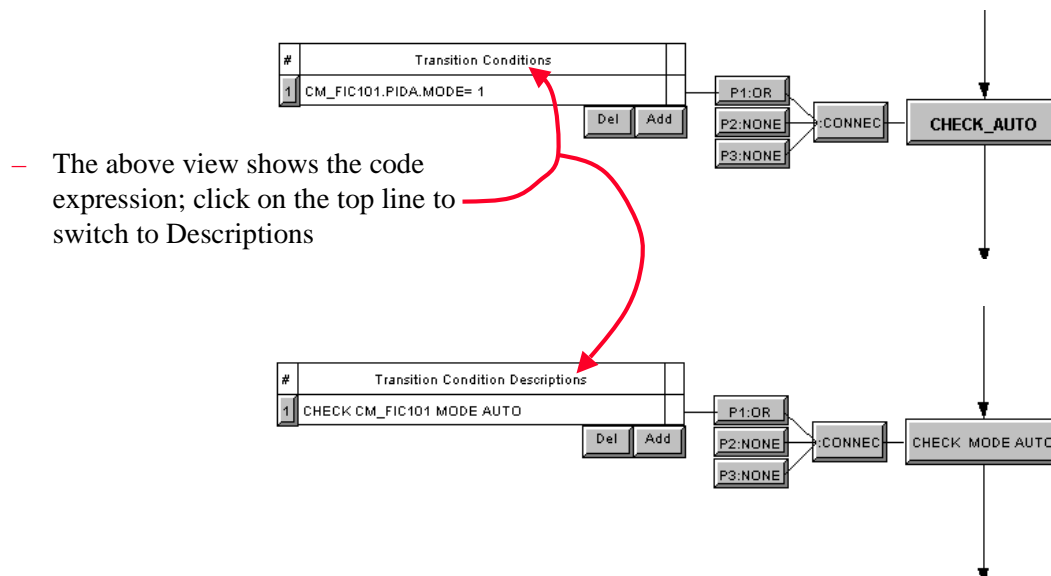
Primary and Secondary Gates


Control Builder has error checking built in. If the Gate choices do not match with the Condition statement layout, Control builder will require you to correct the Gate configuration.



Creating Transitions

- The **Transition** is now configured and appears as shown



- Note: Configuring **Steps** is the same except that there are no Gates and the , Assignment target, is the only function used.

8 - 14

Notes



Adding and Configuring a Step

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **INVOKE_MAIN** Transition

Tab	Name	Description
Main	INITIALIZE_A	INITIALIZE A
	Wait Time 0	Active Time 240
	Description	Output Expression
Out #1	TOTALIZER A COMMAND MODE	CM#_ACCA.TOTAL_A.CMDATTR := 0
Out #2	RESET TOTALIZER A	CM#_ACCA.TOTAL_A.COMMAND := 3
Out #3	SET A TARGET AMOUNT	CM#_ACCA.TOTAL_A.ACCTV := SCM#_XFERA.RECTARGET[1]
Out #4	CM#_FV101 MODEATTR TO PGM	CM#_FV101.DEVCTLA.MODEATTR := 2
Out #5	CM#_PMP101 MODEATTR TO PGM	CM#_PMP101.DEVCTLA.MODEATTR := 2
Out #6	STOP TOTALIZER	CM#_ACCA.TOTAL_A.COMMAND := 2

- Click **OK**
- Wire the **INVOKE_MAIN** Transition to the **INITIALIZE_A** Step

8 - 15

Notes

Outputs to Totalizers

In order for an SCM to command a totalizer, the command attribute of the totalizer must be set to OPERATOR.

Here we Stop and Reset the totalizer, after setting its command attribute to OPERATOR. We then use SCM#_XFERA's first recipe value to store the required charge amount.



For more information on how to configure and use TOTALIZER function blocks refer to the *Control Builder Components Reference, Auxiliary Function, TOTALIZER Block*.



- | Tab | Name | Description |
|--------------|----------------------------|--|
| Main | CHECK_TOTALIZER | CHECK TOTALIZER |
| | Description | Condition Expression Gate |
| Condition #1 | VERIFY TOTALIZER STOPPED | CM#_ACCA.TOTAL_A.STATE = 0 P1 |
| Condition #2 | VERIFY TOTALIZER RESET | CM#_ACCA.TOTAL_A.PV = 0.0 P1 |
| Condition #3 | VERIFY CM#_PMP101 MODEATTR | CM#_PMP101.DEVCTLA.MODEATTR = 2 P2 |
| Condition #4 | | |
| Gates | Pri Gate (1) | Pri Gate (2) Pri Gate (3) Secondary Gate |
| | AND | CONNECT |

- 8 - 16

Notes



Adding and Configuring a Step

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **CHECK_TOTALIZER** Transition

Tab	Name	Description
Main	OPEN_CM#_FV101	OPEN CM#_FV101
	Wait Time 15	Active Time 240
	Description	Output Expression
Out #1	START TOTALIZER	CM#_ACCA.TOTAL_A.COMMAND := 1
Out #2	OPEN CM#_FV101	CM#_FV101.DEVCTLA.GOP := 5
Out #3		
Out #4		

- Click **OK**
- Wire the **CHECK_TOTALIZER** Transition to the **OPEN_CM#_FV101** Step

8 - 17

Notes

Controlling a Device Control Block

The parameter used to control a device control block from an SCM is **GOP**, Generic Output. Its counterpart for monitoring status is **GPV**, Generic Process Variable.

As we saw when we configured device control blocks, the states of a two-state device have ordinal values of 0 and 1. Those of a three-state have 0, 1, and 2. The ordinal values cannot be referenced directly from the SCM as they represent enumerated states.

That is why the Generic parameters were developed. If you look up GOP and GPV in the Knowledge Builder Parameter Reference, you will see that the GOP and GPV value that corresponds to ordinal 0 is 4. For ordinal 1, it is 5 and for ordinal 2, it is 6.

Here we set GOP for **CM#_FV101** to 5, corresponding to ordinal 1, which opens the valve.



For more information on how to configure and use Device Control function blocks refer to the *Control Builder Parameter Reference, GOP / GPV*.



- | Tab | Name | Description |
|------------|----------------------|---|
| Main | CK_FV101_OPEN | CHECK FV101 OPEN |
| | Description | Condition Expression Gate |
| | Condition #1 | CHECK CM#_FV101 OPEN CM#_FV101.DEVCTLA.GPV = 5 P1 |
| | Condition #2 | VERIFY TOTALIZER RUNNING CM#_ACCA.TOTAL_A.STATE = 1 P1 |
| | Condition #3 | |
| | Condition #4 | |
| | Gates | Pri Gate (1) |
| AND | | CONNECT |

- 8 - 18

Notes

[illegible]



Adding and Configuring a Step

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **CHECK_FV101_OPEN** Transition

Tab	Name	Description
Main	OPEN _FV101RC	OPEN FV101RC
	Wait Time 5	Active Time 240
	Description	Output Expression
Out #1	CM#_ FV101RC MODEATTR TO PGM	CM#_ FV101RC.PIDA.MODEATTR := 2
Out #2	CM#_ FV101RC MODE TO MAN	CM#_ FV101RC.PIDA.MODE := 0
Out #3	CM#_ FV101RC OP TO 50%	CM#_ FV101RC.PIDA.OP := 50.0
Out #4	START CM#_PMP101	CM#_PMP101.DEVCTLA.GOP := 5

- Click **OK**
- Wire the **CHECK_FV101_OPEN** Transition to the **OPEN_CM#_FV101RC** Step

8 - 19

Notes

Step Times

Each Step can have two time values configured, the Min Wait Time and the Max Active Time. Each has a different purpose.

The wait time is used to add a specific time period to the process. It is expressed in SCM *cycles*. For example, if the process requires a 10 second delay and the SCM was running on a half second cycle, you would enter a wait time of 20. The Step would then take its configured actions and then start the timer. After 10 seconds, the next Transition would become active.

The active time is used to set an alarm if the SCM is stuck at a particular step. It is also expressed in SCM cycles. If the step is active for longer than the specified active time, the SCM will cause a Time Out process alarm to sound.



For more information on how to configure and use Steps refer to the *Control Builder Components Theory, Sequential Control, STEP Block*.



Adding and Configuring a Transition

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **OPEN_CM#_FV101RC** Step

Tab	Name		Description	
Main	CK_TRANSFER		CHECK TRANSFER	
	Description		Condition Expression	Gate
Condition #1	CHECK CM#_PMP101 ON		CM#_PMP101.DEVCTLA.GPV = 5	P1
Condition #2	CHECK CM#_FV101RC FLOW		CM#_FV101RC.PIDA.PV > 35.0	P2
Condition #3				
Condition #4				
Gates	Pri Gate (1)	Pri Gate (2)	Pri Gate (3)	Secondary Gate
	CONNECT	CONNECT		AND

- Click **OK**
- Wire the **OPEN_CM#_FV101RC** Step to the **CHECK_TRANSFER** Transition

8 - 20

Notes

Transitions to Verify Step Outputs

Through out the SCM we use transitions to verify that step outputs have taken effect. This ensures that the SCM is doing its intended job.



Adding and Configuring a Step

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **CHECK_TRANSFER** Transition

Tab	Name	Description
Main	REGULATE_FLOW	REGULATE FLOW
	Wait Time 0	Active Time 240
	Description	Output Expression
Out #1	CM#_FV101RC MODE TO AUTO	CM#_FV101RC.PIDA.MODE := 1
Out #2	SET FLOW RATE	CM#_FV101RC.PIDA.SP :=SCM#_XFERA. RECTARGET[2]
Out #3		
Out #4		

- Click **OK**
- Wire the **CHECK_TRANSFER** Transition to the **REGULATE_FLOW** Step

8 - 21

Notes

Recipe Value 2

To allow flexibility in the flow of Ingredient A, we are using the second **SCM#_XFERA** recipe parameter to store the flow set point. An operator or another SCM can therefore change the flow rate as required.

Note: To avoid errors in configuring this Step, add the second recipe value prior to using it in the expression. (See page 8-29.)



Adding and Configuring a Transition

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **REGULATE_FLOW** Step

Tab	Name	Description
Main	WAIT_TOTALIZER	WAIT TOTALIZER
	Description	Condition Expression Gate
	Condition #1	WAIT FOR TOTALIZER CM#_ACCA.TOTAL_A.ACCTVFL = 1 P1
	Condition #2	
	Condition #3	
Condition #4		
Gates	Pri Gate (1)	Pri Gate (2) Pri Gate (3) Secondary Gate
	CONNECT	CONNECT

- Click **OK**
- Wire the **REGULATE_FLOW** Step to the **WAIT_TOTALIZER** Transition

8 - 22

Notes

Totalizer Functionality

The totalizer sets a flag when it reaches its target amount. This flag can be used in SCMs or in interlocks to close valves, stop pumps, etc.

The parameter used to monitor this flag is ACCTVFL



For more information on how to configure and use TOTALIZER function blocks refer to the *Control Builder Components Theory, Auxiliary Function, TOTALIZER Block*.



Adding and Configuring a Step

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **WAIT_TOTALIZER** Transition

Tab	Name		Description	
Main	STOP_PUMP		STOP PUMP	
	Wait Time	0	Active Time	240
	Description		Output Expression	
Out #1	ENSURE PMP MODEATTR IS PGM		CM#_PMP101.DEVCTLA.MODEATTR := 2	
Out #2	STOP PUMP		CM#_PMP101.DEVCTLA.GOP := 4	
Out #3				
Out #4				

- Click **OK**
- Wire the **WAIT_TOTALIZER** Transition to the **STOP_PUMP** Step

8 - 23

Notes

Interlocks vs SCM Outputs

CM#_PMP101 is interlocked to stop when **CM#_FV101** is closed or when **CM#_FV101RC** is closed. If the configured interlocks meet the process requirements, then no SCM output is necessary. The action will be handled by the interlock.

Here we are stopping the pump prior to the valves closing and therefore we include the step output to stop the pump.

In the upcoming SCM to control Tank B transfer and Reactor draining, we will allow process overrides to control the corresponding pumps.



Adding and Configuring a Transition

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **STOP_PUMP** Step

Tab	Name	Description	
Main	CHECK_PUMP_OFF	CHECK PUMP OFF	
	Description	Condition Expression	Gate
Condition #1	CHECK CM#_PMP101 IS OFF	CM#_PMP101.DEVCTLA.GPV = 4	P1
Condition #2			
Condition #3			
Condition #4			
Gates	Pri Gate (1)	Pri Gate (2)	Secondary Gate
	CONNECT		CONNECT

- Click **OK**
- Wire the **STOP_PUMP** Step to the **CHECK_PUMP_OFF** Transition

8 - 24

Notes



Adding and Configuring a Step

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **CHECK_PUMP_OFF** Transition

Tab	Name	Description
Main	CLOSE_RCVLV	CLOSE CM#_RCVLV
	Wait Time 0	Active Time 240
	Description	Output Expression
Out #1	CM#_FV101RC MODEATTR PGM	
Out #2	CM#_FV101RC MODE TO MANUAL	
Out #3	CM#_FV101RC OP TO ZERO	
Out #4	CM#_FV101RC MODEATTR TO OPER	

- Click **OK**
- Wire the **CHECK_PUMP_OFF** Transition to the **CLOSE_CM#_FV101RC** Step

8 - 25

Notes

SCMs

From this point forward you will only be given the Descriptions, you will configure the Expression. Solutions are on page 28.



Adding and Configuring a Transition

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **CLOSE_FV101RC** Step

Tab	Name		Description	
Main	CK_RCVLV_CLSD		CK CM#_FV101RC CLSD	
	Description		Condition Expression	Gate
Condition #1	CHECK CM#_FV101RC CLOSED		(Ensure the PV is <= 3)	P1
Condition #2				
Condition #3				
Condition #4				
Gates	Pri Gate (1)	Pri Gate (2)	Pri Gate (3)	Secondary Gate
	CONNECT			CONNECT

- Click **OK**
- Wire the **CLOSE_CM#_FV101RC** Step to the **CHECK_RCVLV_CLSD** Transition

8 - 26

Notes



- | Tab | Name | Description |
|--------|----------------------------|------------------------|
| Main | CLOSE_CM#_FV101 | CLOSE CM#_FV101 |
| | Wait Time 0 | Active Time 240 |
| | Description | Output Expression |
| Out #1 | ENSURE MODEATTR PGM | |
| Out #2 | CLOSE CM#_FV101 | |
| Out #3 | | |
| Out #4 | | |

- 8 - 27

This image shows a single page of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page, leaving small margins at the top and bottom. There is no handwriting or other markings on the paper.



Adding and Configuring a Transition

- Scroll down in the **SCM#_XFERA** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **CLOSE_CM#_FV101** Step

Tab	Name	Description
Main	CK_FV_CLOSED	CHECK FV101 CLOSED
	Description	Condition Expression Gate
Condition #1	CHECK CM#_FV101 CLOSED	P1
Condition #2		
Condition #3		
Condition #4		
Gates	Pri Gate (1)	Pri Gate (2) Pri Gate (3) Secondary Gate
	CONNECT	CONNECT

- Click **OK**
- Wire the **CLOSE_CM#_FV101 Step** to the **CHECK_FV_CLOSED Transition**

8 - 28

Notes

Solutions to expressions

Step **CLOSE_CM#_FV101RC:CM#_FV101RC.PIDA.MODEATTR := 2**

CM#_FV101RC.PIDA.MODE := 0

CM#_FV101RC.PIDA.OP := 0.0

CM#_FV101RC.PIDA.MODEATTR := 1

Transition **CK_CM#_FV101RC_CLSD: CM#_FV101RC.PIDA.PV <= 3.0**

Step **CLOSE_CM#_FV101: CM#_FV101.DEVCTLA.MODEATTR := 2**

CM#_FV101.DEVCTLA.GOP := 4

Transition **CK_CM#_FV101_CLOSED: CM#_FV101.DEVCTLA.GPV = 4**



Configuring Recipe Values

- Double click on empty space to call up the SCM Configuration Screen; click on the **Recipes** tab
- Enter the following information on the **Recipes** tab*

	PARAMETER DESCRIPTION	Value	HI	LO
– 1)	TOTAL_A TARGET	90	150	10
– 2)	CM#_FV101RC FLOW TARGET	80	100	30
- **Close** the SCM and **Save** changes
- Load and Activate **SCM#_XFERA**
- Open Station and configure **SCM#_XFERA** into Group #3 Slot 4
- Start **SCM#_XFERA** by turning on Flag **XFER_A** (Group #2 or #4, **CM#_FLAGS**)
- Monitor from Group #3 and from Control Builder as Ingredient A is transferred to the reactor. Note the alarm resulting from Step **OPEN_FV101RC**. What is its cause? How can it be fixed?

8 - 29

Notes

Configuring Recipe Values

We use Recipe Values #1 and #2 in Steps **INITIALIZE_A** and **REGULATE_FLOW** respectively. Here we configure those Recipe values. To configure Recipe values, you right click to add values. You then highlight the value you wish to input which places its parameters in the parameter entry port.

* As you type in the data, use the <Tab> key to move between entry ports, and to enter the completed value into the list. Only press the <Enter> key when you are finished with all Recipe Value configuration.

Note that we are entering initial values. These can be modified on-line by operators, SCMs, and Batch Packages.

Honeywell

This completes....

PlantScape Controller Implementation

Lesson 1

**Creating an SCM
(SCM#_XFERA)**

8 - 30

Notes

Honeywell

PlantScape Controller Implementation

Lesson 2

Configure and Use Confirm Messages in SCMs

8 - 31

Notes

Introduction

The purpose of this Lesson is to give you the knowledge to be able to configure and use Confirm Messages. Upon completion of this Lesson you will have configured a Confirm Message which will, during SCM#_XFERA execution, prompt the operator to enter the target amount of Ingredient A; cause SCM#_XFERA to wait for this action to complete; and proceed with SCM execution after confirmation.

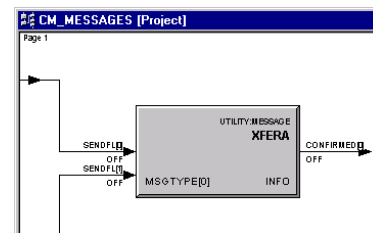
Objectives

- ❶ Add a new message to CM#_MESSAGES
- ❷ Configure this new message to be CONFIRM
- ❸ Modify SCM#_XFERA to utilize this new message



Create a New Message

- Open **CM#_MESSAGES**
- Add the following message as Message Index 2 to the block named **XFERA**: ENTER AMT OF ING A TO SCM#_XFERA RCIPE VAL 1;CNFM WHEN DONE (Note: This is a 60 character message which is the maximum size allowed.)
- Click on the **Message Type** for Message 2 and select CONFIRM.



UTILITY:MESSAGE Block. XFERA - Parameters [Project]

Main | Block Pins | Configuration Parameters | Monitoring Parameters

Name: XFERA

Execution Order in CM: 20

	Message Type	Message Text
0	INFO	FV101 SAFETY INTE
1	INFO	FV101RC SAFETY IN
2	INFO	ENTER AMT OF ING
3	INFO	
4	INFO	

	Message Type	Message Text
0	INFO	FV101 SAFETY INTE
1	INFO	FV101RC SAFETY IN
2	INFO	ENTER AMT OF ING
3	INFO	
4	CONFIRM	
5	INFO	
6	INFO	
7	INFO	

(Note: It is not necessary to add SENDFL[2] as an input pin. We will trigger the message send flag from the SCM.)

- Close and save **CM#_MESSAGES**

8 - 32

Notes

Confirm Messages


Previously, you configured messages that were sent to the Server's Message Summary when safety interlocks occurred. Those messages were information only. Once acknowledged, they disappeared from the Message Summary.

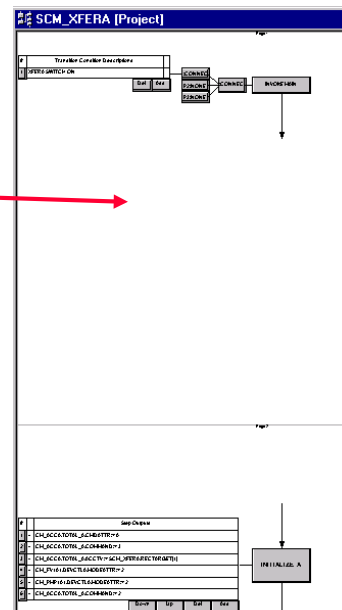
Confirm Messages look and act differently in the Message Summary. They must be Acknowledged and then Confirmed to clear them. They also have a Confirm Flag which can be monitored in an SCM to allow continuation after confirmation.

The difference in configuration is the Message Type choice in the Message Block Main Configuration page.



Use Confirm Message in SCM

- From the **Project** Tree View, double-click **SCM#_XFERA** to open it
- Click on the soft wire from the Invoke Transition to the first Step, and delete it
- In the resulting space, insert a new step. 
- Name the Step **Ing_A_Amt**. Add one command and configure it to send the message created on the previous page to the Station. (Solution located at end of module.)
- Wire the Invoke Transition to Step **Ing_A_Amt**



8 - 33

Notes

Sending Messages from SCMs

An SCM can send a message to the Server's Message Summary by turning on the Send Flag for a message contained in a Message Block. The message can be information only (INFO) or CONFIRM. The Send Flag works the same for each.

Here we are sending a confirm message to prompt the operator to enter a target amount of ingredient A.



Use Confirm Message in SCM (continued)

- Below Step **Ing_A_Amt** add a Transition.
- Name the Transition **Wait_Messg_Cfm**
- Add one Condition and configure it to wait for the operator to confirm the message sent in the last Step (Solution located at end of module.)
- Wire the **Wait_Messg_Cfm** Transition to the **Ing_A_Amt** Step above it
- Wire the **Wait_Messg_Cfm** Transition to the **INITIALIZE** Step below it

8 - 34

Notes

Confirm Messages and SCMs

The SCM can be configured to wait for Message Confirmation. The parameter used in a Transition Condition is the confirm flag, **CONFIRMED**[*n*], where *n* is the message number. The flag goes true when the message is confirmed from station or Control Builder.

Here we are using a Transition Condition to cause **SCM#_XFERA** to wait for the status of **CM#_MESSAGES.XFERA.CONFIRMED**[2] to go true.



Test the Confirm Message

- Inactivate **CM#_MESSAGES** and **SCM#_XFERA**, Download, and Reactivate them
- Open Station and call up Group 3. Start **SCM#_XFERA**
- Click on the Message indicator that flashes green soon after the SCM goes to **Running** status
- Select the Ingredient A Message and Acknowledge it. **Do not Confirm the message!**
- Call up the detail of **SCM#_XFERA** and go to the Recipe page. Enter the desired target amount of Ingredient A

8 - 35

Notes

Operating with Confirm Messages

Confirm messages require two Acknowledgements. The first Acknowledges the message and the second performs the actual Confirmation. The intent is for you to code your SCM so that it waits for the confirmation in order to proceed. In this lesson, we modified **SCM#_XFERA** to make use of this functionality.

The above operation demonstrates how you operate Confirm messages from station.



Test the Confirm Message (continued)

- Navigate back to the Message Summary. Select the Ingredient A Message and Acknowledge it again to Confirm it (Note that it disappears from the summary.)
- The SCM will now continue. Note that the Tank A target amount gets set to the Recipe target you entered previously

8 - 36

Notes

Running SCM#_XFERA

From now on, each time you run **SCM#_XFERA**, you will be prompted to enter a target amount of Ingredient A. If you wish to use the amount already entered, you can confirm the message without a recipe change.



Solutions

Tab	Name	Description
Main	Ing_A_Amt	Ing A Amount
	Wait Time 0	Active Time 240
	Description	Output Expression
Out #1	SEND CONFIRM MESSAGE FOR A AMT	CM#_MESSAGES.XFERA.SENDFL[2] := 1
Out #2		
Out #3		
Out #4		

Tab	Name	Description
Main	WAIT_MESSG_CFM	WAIT MESSG CFM
	Description	Condition Expression Gate
Condition #1	WAIT FOR CONFIRM	CM#_MESSAGES.XFERA .CONFIRMED[2] P1
Condition #2		
Condition #3		
Condition #4		
	Pri Gate (1) Pri Gate (2) Pri Gate (3) Secondary Gate	
Gates	CONNECT NONE NONE CONNECT	

8 - 37

Notes

Solutions to Step and Transition

Your Step and Transition code should be the same as above. Your descriptions can be what ever you choose.

Honeywell

This completes....

PlantScape Controller Implementation

Lesson 2

**Configure and Use Confirm
Messages in SCMs**

8 - 38

Notes

Honeywell

PlantScape Controller Implementation

Lesson 3

Configuring a Supervisory SCM (SCM#_REACTR)

8 - 39

Notes

Introduction

The purpose of this Lesson is to give you the knowledge to be able to configure a Supervisory Sequential Control Module. Upon completion of this Lesson you will have Created a Supervisory SCM that will control other SCMs .

Objectives

- ❶ Create a New SCM
- ❷ Configure Steps and Transition
- ❸ Branch the SCM
- ❹ Use a Flag to allow automatic starting of the SCM



Create a New SCM

- Set up Control Builder with both the **Library** and **Project** views visible
- Click and expand the SYSTEM Library under the **Library** Tree View
- Drag a new SCM from the **Library** to the **Project** view
- Name the SCM **SCM#_REACTR**
- Right-click on the SCM, and click on **Configure Module Parameters...**
- Enter the following information on the Main tab
 - Description **REACTION SUPERVISORY**
 - Keyword **REACTION**
 - Execution Cycle **Default**
 - Execution Phase **-1**
- Close SCM and Assign to **CEE0101**

8 - 40

Notes

SCM_REACTOR Process Description

The SCM#_REACTR controls all other SCMs and CMs to make a complete batch.

In configuring a process it is helpful to flow chart the entire process. By doing this you will obtain an overall view of what you are trying to achieve.

The reactor Process is as follows:

- A Specified amount of Ingredient A is added to the Reactor utilizing the **SCM#_XFERA**. The Agitator is switched on to low speed when the level reaches a specified minimum
- After Ingredient A finishes, you will have the option of adding an amount of Ingredient B manually to the Reactor. The **SCM#_REACTR** is paused during this manual transfer, the agitator continues on low speed
- After Ingredient B finishes the Reactor is heated and cooled down (with the agitator on high speed) using the **SCM#_TEMP**
- The Reactor is drained with the agitator on low speed
- The amounts of Ingredient A, B, and product are logged

Configuring a Supervisory SCM (SCM#_REACTR)**Honeywell****Start Conditions**

- From the **Project** Tree View, double-click the **SCM** to open it
- Configure the following information on the Invoke **Transition**

Tab	Name		Description	
Main	INVOKE_MAIN		INVOKE MAIN	
	Description		Condition Expression	Gate
Condition #1	MANUAL START FLAG		CM#_FLAGS.REACTION.PVFL = 1	P1
Condition #2				
Condition #3				
Condition #4				
Gates	Pri Gate (1)	Pri Gate (2)	Pri Gate (3)	Secondary Gate
	CONNECT			CONNECT

- Click **OK**

8 - 41

Notes**Start Up**

An SCM can be commanded to start through Station. Here we add another method of starting the SCM, namely that if Flag **REACTION** is turned on, the SCM will start from the Idle State.

In **CM#_FLAGS** we configured a block called **REACTION**. The code above looks to that CM to see if the flag is On (1) or Off (0). The flag in the Off state will not keep us from being able to start the SCM with a Start command in Station; it just gives us the option of using the flag to start the SCM as well.

Configuring a Supervisory SCM (SCM#_REACTR)**Honeywell****Adding and Configuring a Step**

- Scroll down in the **REACTR_SCM** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **START_UP** Transition

Tab	Name		Description	
Main	INITIALIZE		INITIALIZE	
	Wait Time	5	Active Time	240
	Description		Output Expression	
Out #1	COMMAND SCM#_XFERA TO START		SCM#_XFERA.COMMAND := 2	
Out #2	RESET REACTOR TOTALIZER		CM#_ACCA.TOTAL_REACTR.COMMAND := 3	

- Click **OK**
- Wire the **START_UP Transition** to the **INITIALIZE Step**

8 - 42

Notes**Starting SCM#_XFERA**

In our Process the first thing we want to do is start the XFERA_SCM. In the first Step Output we give the command

SCM#_XFERA.COMMAND := 2

This statement tells the SCM to change it's state to 2 (Reference Page 7-23, SCM Commands).

Configuring a Supervisory SCM (SCM#_REACTR)**Honeywell****Adding and Configuring a Transition**

- Scroll down in the **REACTR_SCM** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **INITIALIZE** Step

Tab	Name	Description
Main	WAIT_MIN_LEVEL	WAIT MIN LEVEL
	Description	Condition Expression Gate
	WAIT FOR MIN LEVEL	CM#_LVLA.LEVEL_REACTR.PV >= 25.0 P1
	Condition #2	
	Condition #3	
Gates	Pri Gate (1)	Pri Gate (2) Pri Gate (3) Secondary Gate
	CONNECT	CONNECT

- Click **OK**
- Wire the **INITIALIZE Step** to the **WAIT_MIN_LEVEL** Transition

8 - 43

Notes**Waiting For a Minimum Level**

In this Transition we are telling the SCM#_REACTION to wait here until 25 Gallons of ingredient A has been transferred to the Reactor. In our process we do not start the Agitator in an empty Reactor, therefore we code the SCM to wait for a desired amount of product.

Configuring a Supervisory SCM (SCM#_REACTR)**Honeywell****Adding and Configuring a Step**

- Scroll down to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **WAIT_MIN_LEVEL** Transition

Tab	Name	Description
Main	START_AGITATOR	START_AGITATOR
	Wait Time	0
	Active Time	240
	Description	Output Expression
Out #1	AGIT MODEATTR TO PGM	CM#_AGIT101.DEVCTLA.MODEATTR := 2
Out #2	START_AGITATOR LOW	CM#_AGIT101.DEVCTLA.GOP := 4
Out #3	CM#_PMP103 TO PROGRAM	CM#_PMP103.DEVCTLA.MODEATTR := 2
Out #4	CM#_FV103 TO PROGRAM	CM#_FV103.DEVCTLA.MODEATTR := 2

- Click **OK**
- Wire the **WAIT_MIN_LEVEL** Transition to the **START_AGITATOR** Step

8 - 44

Notes**Starting the Agitator**

Our Agitator is controlled by a Device Control Block. To start it we will need to place the Device Control Block in Program Mode Attribute.

CM#_AGIT101.DEVCTLA.MODEATTR := 2

This command tells the Device Control Block to change to Program Mode Attribute. All CMs and SCMs must be in Program Mode Attribute for an SCM to issue internal commands to it. Our next command is

CM#_AGIT101.DEVCTLA.GOP := 4

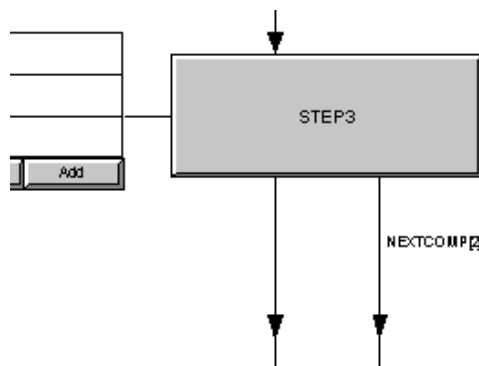
Using this command we are setting the Device Control Block to **S0**(Low).

(Reference 8-17, Device Control Block States GOP/GPV)

Honeywell

Branching

- Right Click on the **START_AGITATOR** Step and add the following pin
 - **NEXTCOMP[2]** Out / Bottom
- We added a Pin to this Step called NEXTCOMP[2]. This Pin gives us the ability to Branch off to a second set of Steps and Transitions. In doing this we can configure two different sets of conditions in our SCM.



8 - 45

Notes

Branching

Branching allows the SCM to take different paths based on different Transition Conditions being true. Here, we are giving the operator the option of interrupting the SCM to manually add Ingredient B if required. Branching always comes from a Step into a Transition. There can be a maximum of 10 branches coming out of a Step.

Branching is not a parallel execution. The branches are evaluated in order. NEXTCOMP[1] to NEXTCOMP[2], and the first one to have true conditions is the branch that is executed.



Branching ...continued

- Branch Conditions (Below is an explanation of the Steps and Transitions to be constructed in the next few pages)
 - The first Branch Transition checks for the following to be true
 - **CM#_AGIT101** is on low
 - **SCM#_XFERA** is complete
 - Checks to see if **ING_B** flag is off (No Ingredient B will be added)
 - The second Branch Transition checks for the following to be true
 - **CM#_AGIT101** is on low
 - **SCM#_XFERA** is complete
 - Checks to see if **ING_B** flag is on (Ingredient B will be added)

8 - 46

Notes

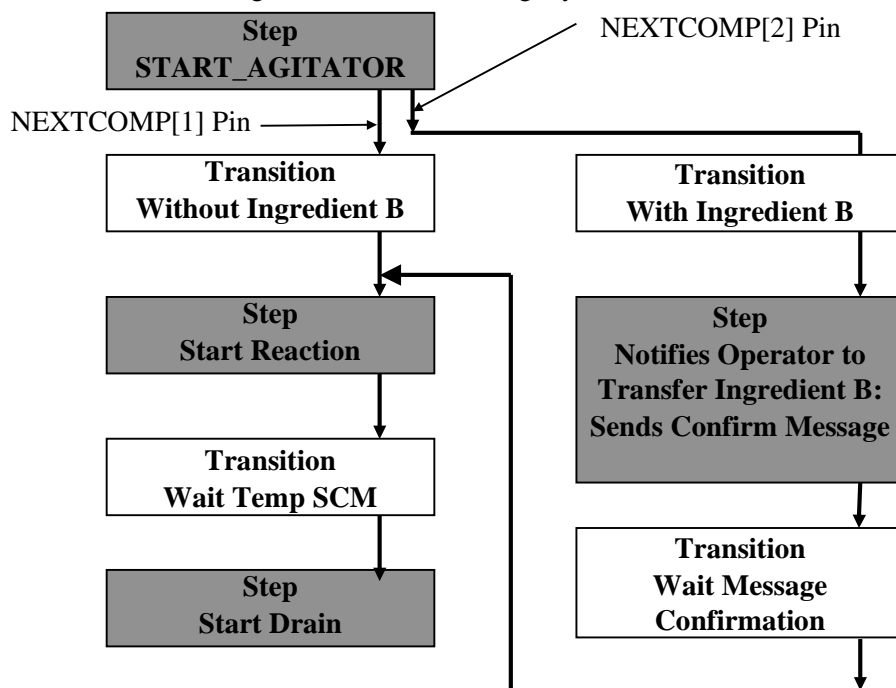
Branching

An SCM can have multiple endings, i.e. branches do not need to come back together. Then the SCM code ends in which ever branch is active, and goes to a completed status.

Honeywell

Branching ...continued

- Below is a drawing of the SCM branching layout



- We will configure the second branch first

8 - 47

Notes

Honeywell

Adding and Configuring a Transition (With Ingredient B)

- Scroll down **and** to the **right** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under and to the right of the **START_AGITATOR** Step

Tab	Name	Description
Main	WITH_ING_B	WITH INGREDIENT B
	Description	Condition Expression Gate
Condition #1	WAIT FOR AGITATOR ON LOW	CM#_AGIT101.DEVCTLA.GPV = 4 P1
Condition #2	WAIT FOR SCM#_XFERA COMPLETE	SCM#_XFERA.STATE = 5 P2
Condition #3	YES ING B (FLAG ON)	CM#_FLAGS.ING_B.PVFL = 1 P3
Condition #4		
Gates	Pri Gate (1)	Pri Gate (2) Pri Gate (3) Secondary Gate
	CONNECT	CONNECT CONNECT AND

- Click **OK**
- Wire the **START_AGITATOR** Step (NEXTCOMP[2] pin) to the **WITH_ING_B** Transition

8 - 48

Notes

With Ingredient B

The with Ingredient B branch will be taken if the ING_B flag is on. This transition will wait for the Agitator to be on low and the SCM#_XFERA to complete. Once both are true and the flag is in the on position

CM#_FLAGS.ING_B.PVFL = 1

the branch will then continue.



Adding and Configuring a Step

- Scroll down to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **WITH_ING_B** Transition

Tab	Name		Description	
Main	MANUALLY_ADD_B		MANUALLY ADD B	
	Wait Time	0	Active Time	240
	Description		Output Expression	
Out #1	MESSAGE TO CHARGE B		CM#_MESSAGES.XFERB.SENDFL[2] := 1	
Out #2				
Out #3				
Out #4				

- Click **OK**
- Wire the **WITH_ING_B** Transition to the **MANUALLY_ADD_B** Step

8 - 49

Notes

Signal to the Operator

Here we are adding a message for the operator. We added a Confirm message to **CM#_MESSAGES** in the **XFERB** block to tell the operator to charge Ingredient B. The operator will manually charge B and then confirm the message.

The SCM waits for the confirmation flag before it proceeds.



Adding and Configuring a Transition

- Scroll down to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **MANUALLY_ADD_B** Step

Tab	Name	Description
Main	MSSG_CNFRM	WAIT MSSG CONFIRM
	Description	Condition Expression Gate
Condition #1	WAIT CONFIRMATION	CM#_MESSAGES.XFERB .CONFIRMED[2] P1
Condition #2		
Condition #3		
Condition #4		
	Pri Gate (1)	Pri Gate (2) Pri Gate (3) Secondary Gate
Gates	CONNECT	CONNECT

- Click **OK**
- Wire the **MANUALLY_ADD_B** Step to the **MSSG_CNFRM** Transition

8 - 50

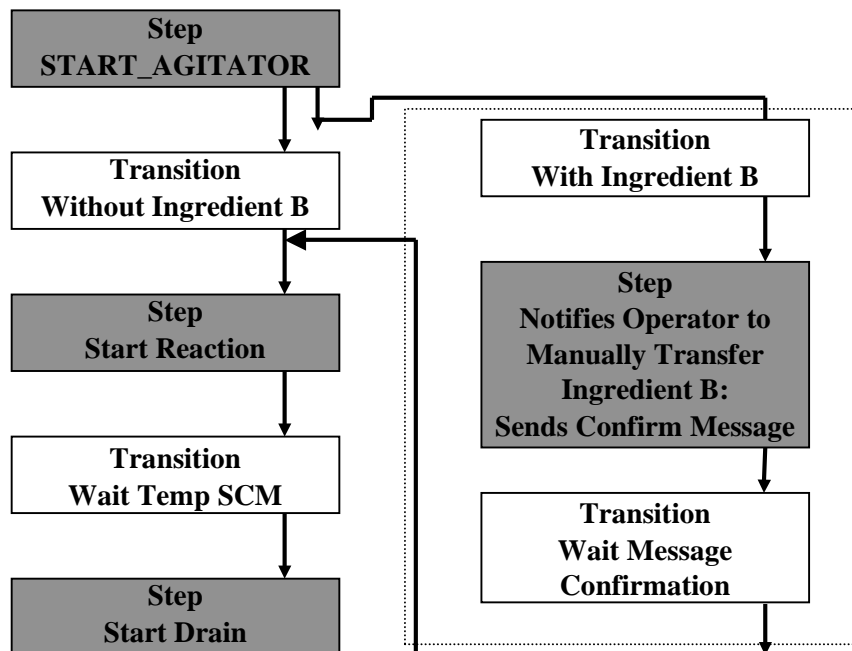
Notes

Required Transition

In order for a Confirm Message to pause an SCM , a Transition condition must wait for the message's Confirm flag to go true. This happens when the message is confirmed in the Station Message Summary page.



- Now you have completed the **with Ingredient B** branch. From this point we will return our configuration to the **without ingredient B** Transition.
- The Without B Transition and Start Reaction Step must be in place to make the wiring connections



8 - 51

Notes

[illegible]

Honeywell

Adding and Configuring a Transition

- Scroll up and **left** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **START_AGITATOR** Step

Tab	Name	Description
Main	NO_ING_B	NO INGREDIENT B
	Description	Condition Expression Gate
Condition #1	WAIT FOR AGITATOR ON LOW	(Solutions starting page 8-59) P1
Condition #2	WAIT FOR SCM#_XFERA COMPLETE	P2
Condition #3	NO INGR B (FLAG OFF)	P3
Condition #4		
	Pri Gate (1)	Pri Gate (2) Pri Gate (3) Secondary Gate
Gates	CONNECT	CONNECT CONNECT AND

- Click **OK**
- Wire the **START_AGITATOR** Step to the **NO_ING_B** Transition

8 - 52

Notes

Without Ingredient B

The without Ingredient B branch will be taken if the ING_B flag is in the off position. This transition will wait for the Agitator to be on low and SCM#_XFERA to complete. Once both are true and the flag is in the off position:

CM#_FLAGS.ING_B.PVFL = 0

the branch will then continue.

Configuring a Supervisory SCM (SCM#_REACTR)**Honeywell****Adding and Configuring a Step**

- Scroll down in the **REACTR_SCM** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **NO_ING_B** Transition

Tab	Name		Description	
Main	START_REACTION		START REACTION	
	Wait Time	0	Active Time	240
	Description		Output Expression	
Out #1	RESET XFERRA			
Out #2	AGITATOR TO HIGH SPEED			
Out #3	START TEMPERATURE SCM			
Out #4				

- Click **OK**
- Wire the **NO_ING_B** Transition to the **START REACTION** Step
- Wire the **MSSG_CNFRM** Transition at the bottom of the second branch to the **START_REACTION** Step

8 - 53

Notes**Start Reaction**

Here we are creating the reaction in our Reactor. Our Step Output #1 is designed to return our Tank A Transfer SCM to the Idle state. This is important in preparing the SCM for later use.

The wiring is more complicated in this step. Here we are making the return connection from our second branch. To better understand how the connection is to be made refer to the diagram on 8-51.



Adding and Configuring a Transition

- Scroll down in the **REACTR_SCM** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **START REACTION Step**

Tab	Name	Description		
Main	WAIT_TEMP	WAIT TEMP		
	Description		Condition Expression	
Condition #1	WAIT_TEMP_SCM COMPLETE			P1
Condition #2	WAIT_TEMP_SCM ABORTED			P1
Condition #3				
Condition #4				
Gates	Pri Gate (1)	Pri Gate (2)	Pri Gate (3)	Secondary Gate
	OR			CONNECT

- Click **OK**
- Wire the **START REACTION Step** to the **WAIT_TEMP Transition**

8 - 54

Notes

TEMP_SCM Complete or Aborted

Here we wait for the SCM#_TEMP to either go to **Completed** (normal operation) or to **Aborted** (abnormal operation). Later we will add an Abort Handler to to the SCM#_TEMP.



Adding and Configuring a Step

- Scroll down in the **REACTR_SCM** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **WAIT_TEMP** Transition

Tab	Name				Description			
Main	START_DRAIN				START DRAIN			
	Wait Time		0		Active Time		240	
	Description				Output Expression			
Out #1	AGITATOR TO LOW							
Out #2	STOP DRAIN TOTALIZER							
Out #3	RESET DRAIN TOTALIZER							
Out #4	SET DRAIN TARGET = REACTR LVL							
Out #5	START DRAIN TOTALIZER							
Out #6	OPEN DRAIN VALVE							

- Click **OK**
- Wire the **WAIT_TEMP Transition** to the **START_DRAIN Step**

8 - 55

Notes

[illegible]

Honeywell

Adding and Configuring a Transition

- Scroll down in the **REACTR_SCM** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **START_DRAIN** Step

Tab	Name	Description
Main	WAIT_DRAIN_OPEN	WAIT DRAIN OPEN
	Description	Condition Expression Gate
Condition #1	WAIT DRAIN OPEN	
Condition #2	WAIT PUMP INTERLOCK CLEAR	
Condition #3		
Condition #4		
Gates	Pri Gate (1)	Pri Gate (2) Pri Gate (3) Secondary Gate
	AND	CONNECT

- Click **OK**
- Wire the **START_DRAIN** Step to the **WAIT_DRAIN_OPEN** Transition

8 - 56

Notes

Timing Issues in SCMs

Often, an SCM will be functional-- it accomplishes the task required-- but there are timing issues to be resolved. Examples include changing the MODEATTR, MODE, and a related parameter on the same CM in the same Step. The commands are executed fast enough that the parameter change command may be executed before the CM is fully in the correct MODE and MODEATTR. The result is a Command Fail alarm which is valid for only one cycle. The fix is to move the parameter change command to the next step.

Here we are avoiding a Command Fail from an interlock, by not only verifying that the drain valve is open prior to turning on the drain pump, but also verifying that the pump interlock is cleared. Without this check, the SCM executes the pump output command in the next Step slightly prior to the interlock clearing, and there is a Fail alarm for one cycle. (Recall that the pump is interlocked OFF when the valve is CLOSED.)



Adding and Configuring a Step

- Scroll down in the **REACTR_SCM** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **WAIT DRAIN OPEN** Transition

Tab	Description			
Main	START_DRN_PUMP		START DRAIN PUMP	
	Wait Time	0	Active Time	240
	Description		Output Expression	
Out #1	START DRAIN PUMP			
Out #2	RESET SCM#_TEMP			
Out #3	STOP AGITATOR			
Out #4				

- Click **OK**
- Wire the **WAIT_DRAIN_OPEN** Transition to the **START_DRN_PUMP** Step

8 - 57

Notes

[illegible]



Adding and Configuring a Transition

- Scroll down in the **REACTR_SCM** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **START_DRN_PUMP** Step

Tab	Name		Description	
Main	WAIT_DRAIN		WAIT_DRAIN	
	Description		Condition Expression	
	Gate			
	Condition #1		WAIT_REACTOR_EMPTY	
	Condition #2			
	Condition #3			
	Condition #4			
Gates	Pri Gate (1)		Pri Gate (2)	
	Pri Gate (3)		Secondary Gate	
	CONNECT		CONNECT	

- Click **OK**
- Wire the **START_DRN_PUMP Step** to the **WAIT_DRAIN Transition**
- Close and save your SCM, then go on to the next module. We will load and operate **SCM#_REACTR** later (after the **Creating Handlers** module.)

8 - 58

Notes

[illegible]

Configuring a Supervisory SCM (SCM#_REACTR)**Honeywell****Code Solutions**

- Transition **NO_ING_B**:
 - Condition 1: CM#_AGIT101.DEVCTLA.GPV = 4
 - Condition 2: SCM#_XFERA.STATE = 5
 - Condition 3: CM#_FLAGS.ING_B.PVFL = 0
- Step **START_REACTION**:
 - Output 1: SCM#_XFERA.COMMAND := 3
 - Output 2: CM#_AGIT101.DEVCTLA.GOP := 5
 - Output 3: SCM#_TEMP.COMMAND := 2
- Transition **WAIT_TEMP** :
 - Condition 1: SCM#_TEMP.STATE = 5
 - Condition 2: SCM#_TEMP.STATE = 17

8 - 59

Notes



Code Solutions

- Step **START_DRAIN**:
 - Output 1: CM#_AGIT101.DEVCTLA.GOP := 4
 - Output 2: CM#_ACCA.TOTAL_REACTR.COMMAND := 2
 - Output 3: CM#_ACCA.TOTAL_REACTR.COMMAND := 3
 - Output 4: CM#_ACCA.TOTAL_REACTR.ACCTV := CM#_LVLA.LEVEL_REACTR.PV
 - Output 5: CM#_ACCA.TOTAL_REACTR.COMMAND := 1
 - Output 6: CM#_FV103.DEVCTLA.GOP := 5
- Transition **WAIT_DRAIN_OPEN** :
 - Condition 1: CM#_FV103.DEVCTLA.GPV = 5
 - Condition 2: CM#_PMP103.DEVCTLA.OI[0] := 0
- Step **START_DRN_PUMP**:
 - Output 1: CM#_PMP103.DEVCTLA.GOP := 5
 - Output 2: SCM#_TEMP.COMMAND := 3
 - Output 3: CM#_AGIT101.DEVCTLA.GOP := 6

8 - 60

Notes

Honeywell**Code Solutions**

- Transition **WAIT_DRAIN** :
 - Condition 1: CM#_LVLA.LEVEL_REACTR.PV \leq 0.0

8 - 61

Notes

Honeywell

This completes....

PlantScape Controller Implementation

Lesson 3

**Configuring a Supervisory SCM
(REACTR_SCM)**

8 - 62

Notes

Honeywell

PlantScape Controller Implementation

Lesson 4

Capturing History Values

8 - 63

Notes

Introduction

The purpose of this Lesson is to give you the knowledge to be able to configure History values. Upon completion of this Lesson you will have configured the History values and set up a step to record the values.

Objectives

- ❶ Configure the History tab
- ❷ Configure a Step to record History values



Capturing History Values

- Open the **SCM#_REACTR**
- Double Click on unoccupied area to open the Parameter Configuration form, and select the **History** Tab
- In the Parameter Descriptor box enter the 3 following **History** settings
 - 1 **TOTAL FROM TANK A**
 - 2 **TOTAL FROM TANK B**
 - 3 **TOTAL FROM REACTION**

(You must Right Click to Append a row. Click in the row and enter the desired text.)

- Click **OK**

Index	Parameter Descriptor	Parameter 1
1	TOTAL FROM TANK A	
2	TOTAL FROM TANK B	
3	TOTAL FROM REACTION	

8 - 64

Notes

Configuring The History Tab

In configuring the History tab all you are doing is assigning names to the different history parameters. The real functionality comes in when you configure a step to actually record the values.



Adding and Configuring a Step (to record amounts)

- Scroll down in the **SCM#_REACTR** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **WAIT_DRAIN** Transition

Tab	Name		Description	
Main	RECORD_AMOUNTS		RECORD AMOUNTS	
	Wait Time	0	Active Time	240
	Description		Output Expression	
Out #1	RECORD A TOTAL		SCM#_REACTR.HISTVALUE[1] := CM#_ACCA.TOTAL_A.PV	
Out #2	RECORD B TOTAL		SCM#_REACTR.HISTVALUE[2] := CM#_ACCA.TOTAL_B.PV	
Out #3	RECORD REACTION TOTAL		SCM#_REACTR.HISTVALUE[3] := CM#_ACCA.TOTAL_REACTR.PV	
Out #4	CLOSE CM#_FV103		CM#_FV103.DEVCTLA.GOP := 4	

- Click **OK**
- Wire the **WAIT_DRAIN** Transition to the **RECORD_AMOUNTS** Step
- Save changes to **SCM#_REACTR**

8 - 65

Notes

Recording Amounts Through a Step

In order for an SCM to record history values, you must configure step outputs to record the amounts into the history values.

Here we are setting **SCM#_REACTR** History Value 1 equal to the total of ingredient A, as recorded by the Ingredient A totalizer. We are setting History Value 2 equal to the Ingredient B total, and History Value 3 equal to the product total.

Note: The values are overwritten each time **SCM#_REACTR** is executed

Honeywell

This completes....

PlantScape Controller Implementation

Lesson 4

Capturing History Values

8 - 66

Notes

Honeywell

PlantScape Controller Implementation

Lesson 5

Creating Handlers

8 - 67

Notes

Introduction

The purpose of this Lesson is to give you the knowledge to be able to configure Handlers. Upon completion of this lesson you will have configured working Check and Abort handlers.

Objectives

- ❶ Configure a Check Handler
- ❷ Configure an Abort Handler



Configuring an Abort Handler

- Open the **SCM#_TEMP**
- From the **Library** view drag a Handler into the **SCM#_TEMP**
- This will bring up the Parameter configuration screen. Enter the following information
 - Name **ABORT**
 - Description **ABORT**

- From the dropdown menu select **ABORT**

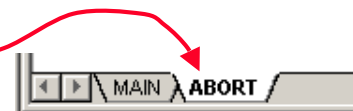
Name:

Description:

Handler Type:

- Click **OK**

- You will see a new tab in the lower left hand corner of the Control Drawing area. This represents the new Handler



8 - 68

Notes

Configuring a Handler

From this selection box you can select any different type of Handler that is available. Every time you create a new Handler a new tab will be added to the bottom of the Control Drawing area. This will enable you to toggle between the different Handlers.



Adding and Configuring a Transition (for the Abort Handler)

- Ensure you have the **ABORT** tab at the bottom of your screen selected
- Configure the following information on the Invoke **Transition**

Tab	Name	Description
Main	INVOKE_ABORT	INVOKE_ABORT
	Description	Condition Expression Gate
	Condition #1	TEMP PV HI HI CM#_TIC101.DACA.PVHHALM.FL =1 P1
	Condition #2	
	Condition #3	
Gates	Pri Gate (1)	Pri Gate (2) Pri Gate (3) Secondary Gate
	CONNECT	CONNECT

- Click **OK**

8 - 69

Notes

Start Conditions

The Abort handler will not interfere with the function of the SCM until the start condition is met. In this case we are working with a Temperature SCM so the Abort Handler will be activated if the High High Alarm is triggered. We have accomplished this by using the condition

CM#_TIC101.DACA.PVHHALM.FL = 1

This condition will hold the Abort Transition in the Invoke position until CM#_TIC101's PV High High alarm is triggered on the DACA block



Adding and Configuring a Step (for the Abort Handler)

- Scroll down in the **ABORT HANDLER** to a fresh screen
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **INVOKE_ABORT** Transition

Tab	Name		Description	
Main	COOL_AND_DRAIN		COOL AND DRAIN	
	Wait Time	0	Active Time	240
	Description		Output Expression	
Out #1	ENSURE MODEATTR FIC101 PROG		CM#_FIC101.PIDA.MODEATTR := 2	
Out #2	CM#_FIC101 MODE TO MAN		CM#_FIC101.PIDA.MODE := 0	
Out #3	SET CM#_FIC101 OP		CM#_FIC101.PIDA.OP := 5.0	
Out #4				

- Click **OK**
- Wire the **INVOKE_ABORT Step** to the **COOL_AND_DRAIN Transition**

8 - 70

Notes

Abort Handler's Content

The next part of this abort will be to configure a Step to resolve our alarm. We have configured our Step to cool the Reactor. The process will be to ensure CM#_FIC101 can be controlled by the SCM and then lower the temperature.



Start Conditions (for the Abort Handler)

- Scroll down in the **ABORT HANDLER** to a fresh screen
- Drag and drop a Transition from the **Library** tab into the SCM control drawing and position it under the **COOL_AND_DRAIN** Transition

Tab	Name		Description	
Main	ABORT_WAIT_TEMP		ABORT_WAIT_TEMP	
	Description		Condition Expression	Gate
Condition #1	WAIT 55 DEG C		CM#_TIC101.PIDA.PV <= 55.0	P1
Condition #2				
Condition #3				
Condition #4				
Gates	Pri Gate (1)	Pri Gate (2)	Pri Gate (3)	Secondary Gate
	CONNECT			CONNECT

- Click **OK**
- Wire the **COOL_AND_DRAIN Step** to the **ABORT_WAIT_TEMP Transition**
- Close and save changes to the **SCM#_TEMP**

8 - 71

Notes



Configuring a Check Handler

- Open the **SCM#_REACTR**
- From the **Library** view drag a Handler into the **SCM#_REACTR**
- This will bring up the Parameter configuration screen. Enter the following information
 - Name **Check**
 - Description **Check**
- From the dropdown menu select **CHECK**
- Click **OK**
- You will see a new tab in the lower left hand corner of the Control Drawing area, this represents the new Handler



8 - 72

Notes

Configuring a Handler

From this selection box you can select any different type of Handler that is available. Every time you create a new Handler a new tab will be added to the bottom of the Control Drawing area.

NOTE: Refer back to the Introduction to SCMs section for review of Handler types.



Adding and Configuring a Step (for the Check Handler)

- Ensure you have the **CHECK** tab at the bottom of your screen selected
- Drag and drop a Step from the **Library** tab into the SCM control drawing and position it under the **Invoke Check** Transition

Tab	Name	Description
Main	SETUP	SETUP
	Wait Time 0	Active Time 240
	Description	Output Expression
Out #1	MODEATTR SCM# XFERRA TO PGM	SCM# XFERRA.MODEATTR := 2
Out #2	MODEATTR SCM# TEMP TO PGM	SCM# TEMP.MODEATTR := 2
Out #3	DRAIN VALVE MODEATTR TO PGM	CM# FV103.DEVCTLA.MODEATTR := 2
Out #4	DRAIN PUMP MODEATTR TO PGM	CM# PMP103.DEVCTLA.MODEATTR := 2
Out #5	DRAIN TOTALIZER TO PGM	CM# ACCA.TOTAL_REACTR.COMDATTR := 0
Out #6	A TOTALIZER TO PROGRAM	CM# ACCA.TOTAL_A.COMDATTR := 0
Out #7	B TOTALIZER TO PROGRAM	CM# ACCA.TOTAL_B.COMDATTR := 0
Out #8	RESET B TOTALIZER	CM# ACCA.TOTAL_B.COMMAND := 3
Out #9	RESET DRAIN TOTALIZER	CM# ACCA.TOTAL_REACTR.COMMAND := 3
Out #10	RESET INGR B SIGNAL	CM# FLAGS.ING_B.PVFL := 0
Out#11	RESET TOTAL_A	CM# ACCA.TOTAL_A.COMMAND := 3

- Click **OK**
- Wire the **Invoke_Main** Transition to the **SETUP** Step

8 - 73

Notes

Configuring a Check Handler

In this Check Handler we are resetting all the totalizers and setting all the components needed to Program mode. This will allow us to run the process problem free.



Loading, Activating and Operation you new SCMs

- Inactivate the **SCM#_TEMP**
- Load and Activate the following SCMs
 - **SCM#_TEMP**
 - **SCM#_REACTR**
- In **Station**, add **SCM#_REACTR** to **Group #4, Slot 7**
- Operate the **SCM#_REACTR** from station with the **ING_B** flag on and off to test branching functionality.
- After it completes click on the **History Values** tab. Note the totals of Ingredient A, B and Product

8 - 74

Notes

Honeywell

This completes....

PlantScape Controller Implementation

Lesson 5

Creating Handlers

8 - 75

Notes

Honeywell

PlantScape Controller Implementation

Lesson 6

Configure a Common SCM (SCM#_XFER) (Optional)

8 - 77

Notes

Introduction

The purpose of this Lesson is to give you the knowledge to be able to configure a Common Sequential Control Module and its Alias Table. Upon completion of this Lesson you will have Created a Common SCM that will control material transfer. Using specific instances from the Alias Table you will apply this SCM to the transfer of ingredient B to the Reactor.

Objectives

- ❶ Create a New SCM
- ❷ Create an Alias Table for the SCM
- ❸ Configure Steps and Transitions using “generic” parameters from the Alias Table
- ❹ Run the SCM using devices for the transfer of Ingredient B



Introduction

- A **Common SCM** is one that can control several equipment units, one at a time, with no code change requirements. The decision on which unit to control can be made at run time through a single parameter choice in the SCM's Alias Table.
- An **Alias Table** is a matrix that associates alias names with the actual parameters that the aliases resolve to at run-time. The Alias Table is the key component to the Common SCM function. It creates the foundation for dynamic indirection
- **Dynamic indirection** allows common SCM Step outputs and Transition conditions to communicate to different CM blocks at run-time. This provides the ability for a user to create a single SCM that may control different equipment each time it runs.
- The Alias Table owner tries to locate the specific CM blocks for the aliases at load time or at run-time. This process is called **binding**. The binding process results in the creation of two sets of connections: the connection between the common SCM Step output or Transition condition code and the aliases; and the connection between the aliases and the referenced CM blocks.
- Binding status is visible in the Monitoring Tab, SCM's Configuration Parameters, **Aliases** tab
- We will now configure a common SCM

8 - 78

Notes

Background

A SCM can be configured to reference aliases instead of actual C200 parameters. The specific C200 parameters to be used when the SCM executes can then be assigned dynamically. Hence, one SCM can control various sets of CMs with out any changes in its configuration.

The mechanism used in the C200 to allow this common code configuration is the Alias Table. It must be configured on the common SCM prior to Step\Transition configuration. Since the Alias Table establishes function block type and parameter for each alias, the standard Control Builder Point Selection and Enumeration Fetch are functional with the "generic" SCM code.

Note: References to aliases defined in other SCMs' Alias Tables are not supported in software version R400



For more information about Common SCMs refer to
Control Builder Components Theory, Common SCMs.



Create a New SCM

- Add a new SCM to your project
- In the Project view, right-click on the SCM, and click on **Configure Module Parameters...**
- Enter the following information on the Main tab
 - Name **SCM#_XFER**
 - Description **COMMON SCM FOR TRANSFER**
 - KEY WORD **XFER**
- Close SCM and Assign to **CEE0101**

8 - 79

Notes

SCM#_XFER Process Description

The SCM#_XFER automates the transfer of material using a totalizer, a 2-state valve, and a 2-state pump.

The transfer process is as follows:

- A Specified amount of material is entered into the SCM's Recipe Target Value 1.
- MODEATTR parameters for both the valve and the pump are set to Program
- The totalizer target is set equal to the Recipe Target Value. The valve is then opened
- After the pump interlock clears, the pump is turned on.
- When the totalizer reaches the target amount, interlocks close the valve which in turn interlocks the pump off.
- The modeattr parameters of the pump and drain are then set back to Operator.
- The SCM notifies the Server Message Summary of the completed material transfer and then goes to Complete status

Honeywell

Alias Table

- First, we must configure the Alias Table
- Right-click on the SCM, and click on **Configure Module Parameters...**
- Select the **Aliases** Tab to bring up the form shown below:

SYSTEM:SCM Block, SCM_XFER - Parameters [Project]

Main | Handlers | Alarm | Recipe | History | Aliases | Server | Status

☐ Enable Alias Configuration

Number of Aliases: 1

Number of Instances: 1

Instance Selected: 1

#	Alias	Model Block	Model Param	Status	Instance 1
1					

Use right mouse button to invoke the edit menu

☐ Show Parameter Names

OK Cancel Help

8 - 80

Notes

Alias Table

A common SCM is added to the Project just like any other. The property that makes it common is the alias table and the use of aliases in the code.

The **Alias** tab is where the table is configured.



Alias Table (continued)

- Click the **Enable Alias Configuration** check box
- Input 11 for the **Number of Aliases** and 1 for the **Number of Instances**

#	Alias	Model Block	Model Param	Status	Instance 1
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					

8 - 81

Notes

Alias Table (continued)

In order to configure an Alias Table, table configuration must first be enabled.

Next the **Number of Aliases** and **Number of Instances** are entered. This sets up the initial table size. Both of these can be changed in the Project Tab at any time. Also, you can right click to copy and paste rows, or to append columns and rows. The maximums are **500 Aliases** and **100 Instances** per table. An additional restriction is that the Number of Aliases times the Number of Instances must be ≤ 4500 .

Here we need 11 parameters to configure the SCM to accomplish the transfer of material. We therefore enter 11 for the **Number of Aliases**. We are doing only Transfer Ingredient B so we need only one **Instance**.

Honeywell

Alias Table (continued)

- Fill out the table to include the following aliases:

#	Alias
1	TOTALIZER_CMD
2	TOTALIZER_STATUS
3	VALVE_MODEATTR
4	VALVE_GOP
5	PUMP_MODEATTR
6	PUMP_GOP
7	VALVE_GPV
8	PUMP_OI_0
9	TOTALIZER_ACCTV
10	TOTALIZER_ACCTVFL
11	END_MESSAGE

- Select the proper Function Block type in the **Model Block** column and the proper parameter in the **Model Param** column to match the Parameters required to accomplish material transfer (Parameters for Ingredient B Transfer are shown on the next page.)

8 - 82

Notes

Alias Table (continued)

Each Alias consists of a name, a block type or **Model Block**, and a parameter type or **Model Param**. The Model Block and Model Param allow Control Builder to verify that the “generic” code is properly configured.

To connect the Aliases to actual C200 parameters, **Instances** are added to the table. The set of parameters to be used at run time can be selected by choosing a particular instance.



Alias Table (continued)

- Enter these parameters for the Transfer B instance. (**Remember to add your # number!**)

Instance 1
CM_ACCA.TOTAL_B.COMMAND
CM_ACCA.TOTAL_B.STATE
CM_FV102.DEVCTLA.MODEATTR
CM_FV102.DEVCTLA.GOP
CM_PMP102.DEVCTLA.MODEATTR
CM_PMP102.DEVCTLA.GOP
CM_FV102.DEVCTLA.GPV
CM_PMP102.DEVCTLA.OI[0]
CM_ACCA.TOTAL_B.ACCTV
CM_ACCA.TOTAL_B.ACCTVFL
CM_MESSAGES.XFERB.SENDFL[1]

- The completed table is shown on the next page

8 - 83

Notes

Instances

Instance Parameters are what tie the Alias Table to C200 parameters. At run time, a particular instance can be chosen.


The instance selection may be changed in the following ways at run-time:

- Change the instance selection from a display
- Store to the Common SCM's parameter **INSTSELEC** from an SCM Step Output
- Store to **INSTSELEC** from Total Plant Batch



Alias Table (continued)

- Below is the completed Alias Table (Connection **Status** will be active in the Monitoring tab.)



#	Alias	Model Block	Model Param	Status	Instance 1
1	TOTALIZER_CMD	AUXILIARY:TOTALIZER	COMMAND		CM_ACCA.TOTAL_B.COMMAND
2	TOTALIZER_STATUS	AUXILIARY:TOTALIZER	STATE		CM_ACCA.TOTAL_B.STATE
3	VALVE_MODEATTR	DEVCTL:DEVCTL	MODEATTR		CM_FV102.DEVCTLA.MODEATTR
4	VALVE_GOP	DEVCTL:DEVCTL	GOP		CM_FV102.DEVCTLA.GOP
5	PUMP_MODEATTR	DEVCTL:DEVCTL	MODEATTR		CM_PMP102.DEVCTLA.MODEATTR
6	PUMP_GOP	DEVCTL:DEVCTL	GOP		CM_PMP102.DEVCTLA.GOP
7	VALVE_GPV	DEVCTL:DEVCTL	GPV		CM_FV102.DEVCTLA.GPV
8	PUMP_OI_0	DEVCTL:DEVCTL	OI		CM_PMP102.DEVCTLA.OI[0]
9	TOTALIZER_ACCTV	AUXILIARY:TOTALIZER	ACCTV		CM_ACCA.TOTAL_B.ACCTV
10	TOTALIZER_ACCTVFL	AUXILIARY:TOTALIZER	ACCTVFL		CM_ACCA.TOTAL_B.ACCTVFL
11	END_MESSAGE	UTILITY:MESSAGE	SENDFL		CM_MESSAGES.XFERB.SENDFL[1]

- Next, we will configure the SCM using the Aliases

8 - 84

Notes

Status Field

The Status column in the Alias Table is blank in the Project view. It indicates the connection status from the Alias to the SCM, and from the Alias to all of its block instances.

Status is established at download or at runtime and is visible in the Monitoring tab.



SCM Configuration with Aliases

- As with all SCMs, we will be coding Steps and Transitions. For this exercise, you will be given a detailed process description from which you will configure the SCM.
 - Step names, Transition names, and descriptions can be anything you choose
 - Step Outputs and Transition Conditions will use the Aliases only
 - The process description is on the next page
 - A complete solution is at the end of the module
 - Below is the table for the first Step to demonstrate code using aliases. Note that the alias is treated as an SCM parameter in the code (SCMname.alias)

Tab	Name	Description
Main	INITIALIZE	INITIALIZE
	Wait Time 0	Active Time 240
	Description	Output Expression
Out #1	VALVE MODEATTR TO PGM	SCM#_XFER.VALVE_MODEATTR:= 2
Out #2	PUMP MODEATTR TO PRM	SCM#_XFER.PUMP_MODEATTR := 2
Out #3	RESET TOTALIZER	SCM#_XFER.TOTALIZER_CMD := 3
Out #4	START TOTALIZER	SCM#_XFER.TOTALIZER_CMD := 1
Out #5	SET TARGET AMT = RECIPE 1	SCM#_XFER.TOTALIZER_ACCTV := SCM#_XFER.RECTARGET[1]

8 - 85

Notes

Common SCM Code

The use of Aliases in SCM code allows it to be applied to different Instances.

Aliases are actually parameters of the SCM in which their Table is configured. In the SCM code, therefore, we reference the Aliases just like any other SCM parameter, namely SCM.Alias.

In R400, aliases are only available locally to the SCM which contains the Alias Table.



Process Description

- Start the SCM on command (no start conditions)
- Ready the process
 - Set the pump and valve MODEATTRs to PROGRAM
 - RESET and START the totalizer
 - Set the totalizer target equal to the SCM Recipe Target Value 1
- Verify the MODEATTR on the valve is PROGRAM and the totalizer is running
- Open the valve
- Verify the valve is open and the pump interlock is clear
- Start the pump
- Wait for the totalizer actual amount to reach the target amount
- Set the pump and valve MODEATTRs to operator
- Send a message stating that SCM is complete

8 - 86

Notes

Process Description

Recall that we configured an INFO type Message at Index 1 in the XFERB Message block in CM#_MESSAGES to indicate that the transfer of ingredient B is complete. Here is where we use that message.



SCM Operation

- After configuration, close and save your SCM
- Download and Activate your SCM
- Look at the Alias Table in Monitoring to check the Connection (binding) Status to the SCM code and the transfer B CMs *
- Add SCM#_XFER to Group 3, Slot 7
- Enter a target amount into SCM#_XFER Recipe Value 1
- Run the SCM

8 - 87

Notes

***Binding Status**

The binding status is represented by parameter STATUS. This parameter is an enumeration with value Null (0), OK (1) and Binding (2). If the data owner/reference destination block of an alias is successfully connected, the STATUS of the alias will be set to OK. If the instance parameter is not configured (null instance parameter), the STATUS of the alias will be set to Null. If the data owner/reference destination block of an alias can not be located or connected, the STATUS of the alias will be set to Binding.

Failure to bind is caused by any of the following situations:

The data owner/reference destination block is not loaded;

The data owner/reference destination block is deleted;

The SCM does not use the alias in the code anywhere;

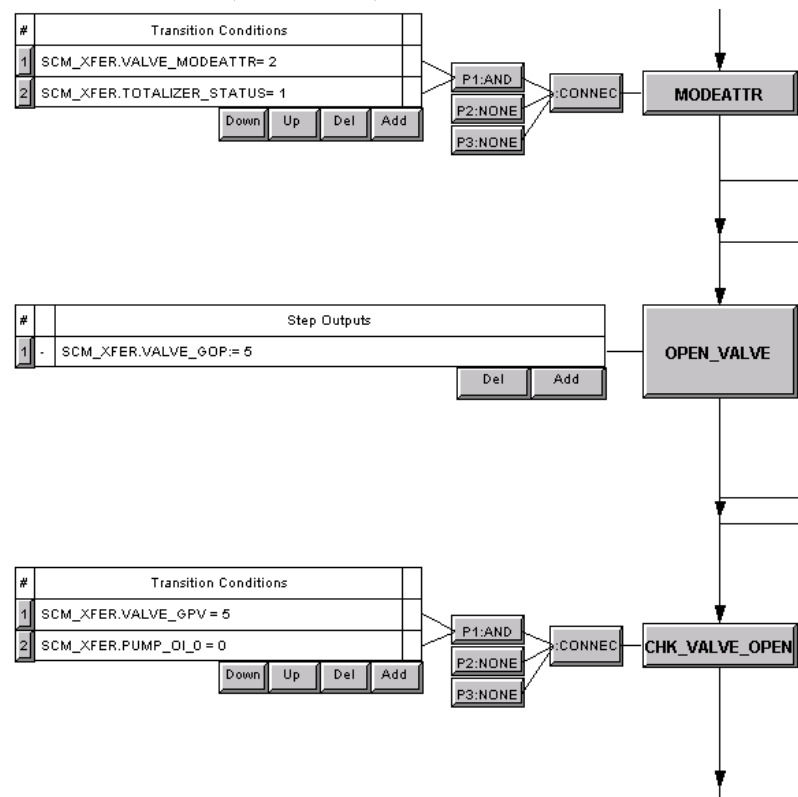
The parameter referenced cannot be continuously connected, for example, the send flag for a message, which is a pulse trigger.

-
- The diagram illustrates the initial state configuration of a state machine. At the top, a table titled "Transition Conditions" has a single row with a column for the condition number (containing "#") and a column for the condition description (containing "Transition Conditions"). To the right of this table is an "Add" button. Below the table, a vertical stack of three boxes labeled "P1:NONE", "P2:NONE", and "P3:NONE" are connected by arrows to a box labeled "S:NONE". This "S:NONE" box is then connected to an "Invoke" button. A vertical arrow points down from the "Invoke" button to an "INITIALIZE" button. To the left of the "INITIALIZE" button is a table titled "Step Outputs" with five rows. The first row contains "1" and "SCM_XFER.VALVE_MODEATTR := 2". The second row contains "2" and "SCM_XFER.PUMP_MODEATTR := 2". The third row contains "3" and "SCM_XFER.TOTALIZER_CMD := 3". The fourth row contains "4" and "SCM_XFER.TOTALIZER_CMD := 1". The fifth row contains "5" and "SCM_XFER.TOTALIZER_ACCTV := SCM_XFER.RECTARGET[1]". To the right of this table are four buttons: "Down", "Up", "Del", and "Add". A vertical arrow points down from the "INITIALIZE" button to the bottom of the diagram.

Notes

Honeywell

SCM#_XFER Possible Solution (continued)



8 - 89

Notes



Notes

Honeywell

This completes....

PlantScape Controller Implementation

Lesson 6

**Configure a Common SCM
(SCM#_XFER)**

8 - 91

Notes

Unit 8 Exam

QuesNo	Question
1	<p>For an SCM start automatically using start conditions, you need to add conditions to the Invoke Transition.</p> <p>A True B False</p>
2	<p>The method used to add Steps and Transition to a Handler is?</p> <p>A Selecting File > New > Step (or Transition) B Right Clicking in the SCM and Selecting New > Step (or Transition) C Dragging the new Step or Transition from the Library view to the Project View. D Dragging the new Step or Transition from the Library view into the open SCM in the Control drawing area.</p>
3	<p>For an SCM to be able to record History values all you need to do is configure the History Tab in the SCM Parameter Configuration form.</p> <p>A True B False</p>
4	<p>Handlers have multiple purposes. In this course we configured a check handler. What is the purpose of a check handler?</p> <p>A To check all the attributes of all the SCM components B Initialize process equipment and/or reset values for a new SCM run C To run the Check-Mode program for process distribution</p>

Unit 9

Import\Export

Honeywell

PlantScape Controller Implementation

Lesson 1

Import\Export (Optional)

9 - 3

Notes

Introduction

The purpose of this Lesson is to give you the knowledge to be able to incorporate parts from one Control Builder Project into another. The method used for this is called Import\Export. It is a tool contained in Control Builder. To try this functionality, you will create a new CM and Export it to text files. You will then Import from a classmate's Project Export files into your project.

Objectives

- ❶ Create a new CM named CM_FV#01 (where # is your team number)
- ❷ Export the new CM to text files
- ❸ Import a classmate's CM_FV#01 into your project

Note: This is the same tool we used to Import SCM#_TEMP in a previous lab. Here you will do both the Export and Import



Background

- Your Project File can contain a great many items. It may contain IOMs, CMs, and SCMs for up to ten C200 controllers
- There is a tool which allows the copying of some or all of a project to insert into another project. That tool is the Import\Export tool.
- Import\Export is accessed from Control Builder
- Export copies the designated portion of the project from Control Builder to text files
- Import copies from the designated text files into Control Builder's Project tab
- Both Import and Export have useful features which we will examine in this module
- First we will examine the Export function

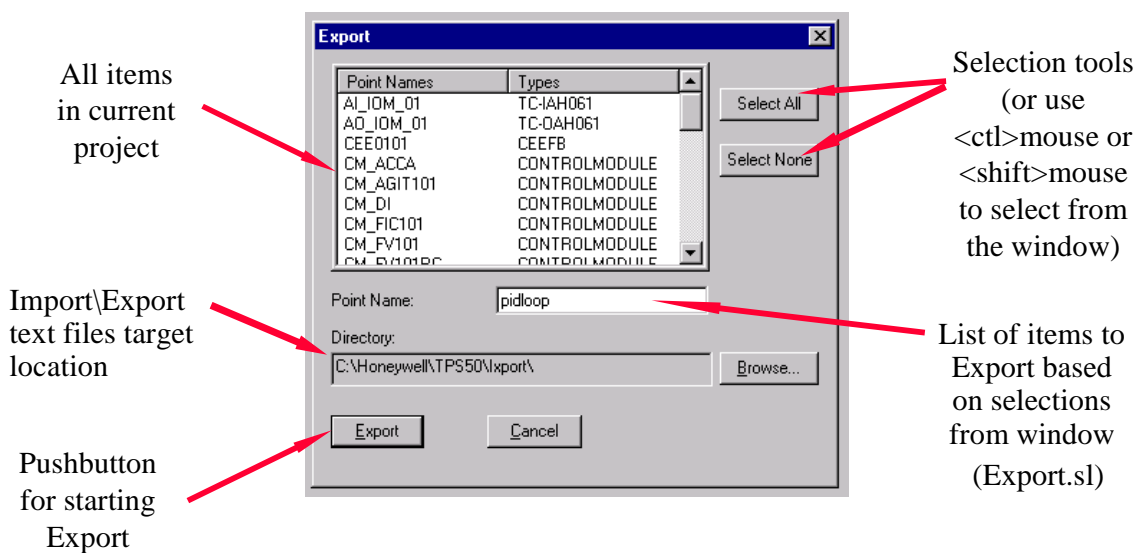
9 - 4

Notes

Honeywell

Export

- In Project, **Copy** CM#_FV101 and name it CM_FV#01
- Select **File, Export** to call up the dialog box shown:



9 - 5

Notes

Export

The Export function copies designated Control Builder Project items to the directory shown in the Export dialog box. The items are converted into text files, one file for each function block along with other files used by the Import\Export function.

An additional file, Export.sl, is created. It is a selection list which contains the tag names of the items chosen to Export.

The default file location for the Export is Honeywell\tps50\Iexport

Honeywell

Export ...continued

- Scroll down in the Project window and find **CM_FV#01** created in the last step
- Select the CM and note that it now appears in the **Point Name** port
- Select the **Export** pushbutton to cause the CM to be put into export format in the target path location.
- Locate the resulting files in windows Explorer. Note that there are two files for each exported item, one with a .cnf.xml extension, the other with a .bcd extension. The last export selection list also appears in a file called Export.sl. Note the small size of the files (Your entire class project will fit on a single floppy in Export format.)
- We will now Import a CM into the Project

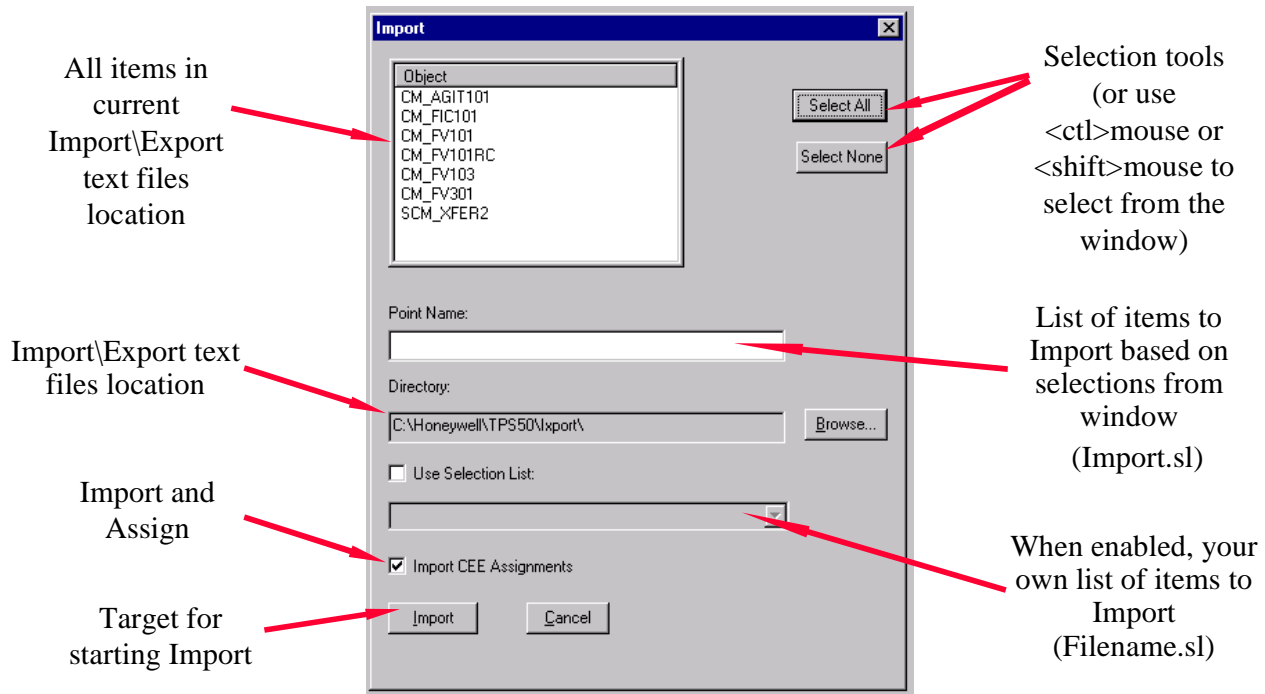
9 - 6

Notes

Honeywell

Import

In Control Builder, select **File, Import** to call up the dialog box shown:



9 - 7

Notes

Import

The Import function copies designated items from the Import\Export text files, located in the directory shown in the Import form, to the Control Builder Project.

The items to be Imported are listed in a file called a selection list. If items are selected from the Import form window, the list created is called Import.sl. You can use notepad to create your own selection list as well. To use this type of list, copy your .sl file to the same directory that contains the item text files, check the Use Selection List check box, and browse to the list file. The items contained in the list will appear in the Point Names port.

If the items were assigned in the source project file, and the same controller name exists in the target project file, assignment to the controller can be done at Import. If this is not desired, do not check the Import CEE Assignments check box. The Imported Items will then appear under the project root.

Honeywell

Import ...continued

- Select the **Browse...** pushbutton and find a classmate's Export Directory. Note the list of items in the Import\Export text files location on that PC
- Select the CM_FV#01 point Exported in the last step. Note that it now appears in the **Point Name** port
- Uncheck the **Import CEE Assignment** check box
- Select the **Import** target
- Notice that the CM is now in your project under the Root, since Assignment was not Imported
- The CM can now be modified, assigned, and loaded to a controller

9 - 8

Notes

Honeywell

This completes....

PlantScape Controller Implementation

Lesson 1

Import\Export

9 - 9

Notes

Book 2

Appendix

Group Configuration

Group #1	Hardware
Slot 1:	CPM0101
Slot 2:	CEE0101
Slot 3:	DI_IOM_01
Slot 4:	DO_IOM_01
Slot 5:	AI_IOM_01
Slot 6:	AO_IOM_01
Slot 7:	DO_IOM_02
Slot 8:	Empty

Group #2	Temperature
Slot 1:	CM#_TIC101
Slot 2:	CM#_FIC101
Slot 3:	CM#_AGIT101
Slot 4:	CM#_FLAGS
Slot 5:	SCM#_TEMP
Slot 6:	Empty
Slot 7:	Empty
Slot 8:	Empty

Group #3 Transfer A & B
Slot 1: CM#_FV101
Slot 2: CM#_FV101RC
Slot 3: CM#_PMP101
Slot 4: SCM#_XFERA
Slot 5: CM#_FV102
Slot 6: CM#_PMP102
Slot 7: SCM#_XFER
Slot 8: CM#_ACCA

Group #4 Drain
Slot 1: CM#_FV103
Slot 2: CM#_PMP103
Slot 3: CM#_AGIT101
Slot 4: CM#_ACCA
Slot 5: CM#_LVLA
Slot 6: CM#_FLAGS
Slot 7: SCM#_REACTR
Slot 8: Empty

Group #5 SCMs (Suggested group to monitor all SCMs – not configured in labs)
Slot 1: SCM#_XFERA
Slot 2: SCM#_TEMP
Slot 3: SCM#_XFER
Slot 4: SCM#_REACTR
Slot 5: CM#_ACCA
Slot 6: CM#_LVLA
Slot 7: CM#_FLAGS
Slot 8: Empty