Honeywell

Lab Exercises

4410S

TotalPlant

Notices and Trademarks

Copyright 1993, 1997, 1999 by Honeywell Inc.
Release 5.0 September 28, 1999

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.

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 85053-3028

1-800 852-3211

LAB #1

BUILD AM POINTS

OBJECTIVE

Using the Blend Tank Simulator Design Specifications, build, load, activate, and test AM Custom, Regulatory, and Flag points. Save and load data to/from an .IDF.

PROCEDURE

Your partition number is: _____

Your Area number is: _____ , Your assigned groups are :_____

The instructor volume is: _____

1. Copy the exception-build file SIMLVL00.EB from the instructor volume to your partition volume under a new filename. Using the Text Editor, modify it to reflect your point names and use it to exception-build your regulatory and flag points to an IDF. Don't load the points yet.

2. Using the Blend Tank Simulator design specification, build the AM Custom point SIMLVLxx, where xx is your student directory, using the PED. Configure the point with a single CL Block and a CDS package, SIMLVCDS, which already resides on the Lab system. Enter the proper entity names into the CDS parameters per Design Specification, and also the following formula data:

Parameter	Value	
TSTRING(1)	BROWN STOK	- formula name
TREAL(1)	022	- formula code
TREAL(2)	0.32	- dye 1 ratio
TREAL(3)	0.70	- dye 2 ratio
TREAL(4)	0.15	- dye 3 ratio
TREAL(5)	30.0	- agitator min. level
TREAL(6)	0.2	- residence time constant

   Write this point to the same IDF you created in step 1, then multiple-load all the points to the AM.

3. Build groups in your assigned area and group number to hold the points just created.

4. Call up and activate your Regulatory points. Check the parameters to make sure they were built properly. Test them by changing modes, setpoints, and outputs.

LAB #1, BUILD AM POINTS Continued

5. Call up the Flag points and check them by changing the PVs.

6. Call up the Custom point and check the parameters. If there are errors, fix them by reconstituting the point and re-loading.

7. Build and load your History Group containing the SP and PV of the Pulp Flow Controller (FICPLPxx) and one Dye Flow Controller (FICDY1xx, FICDY2xx, or FICDY3xx). Your assigned Unit ID and History Group may be found on your Partition sheet.

LAB #2

TEST THE BLEND TANK SIMULATOR

OBJECTIVE

Link a pre-built CL Block to a simulation Custom point. Start-up and test the simulation from a process schematic.

PROCEDURES

1. Copy the pre-built CL Block SIMLVL00.AO to your volume from the instructor volume. Using the full pathname, link the Block to your AM Custom point. The Command Processor command will have the form:
   `LK NET>vol>SIMLVL00  SIMLVLxx` where xx is your student directory.

2. Display the Custom point to make sure that the Block linked properly. Activate the point.

3. Call up the pre-built schematic BLDTNKxx, where xx is your student directory,.at the Operator Station and put the controllers in their normal modes (CAS for the dye flow controllers, AUTO for the pulp flow and level controllers). Select the pump and get it running. Select the 3-way valve and put it in the running position. If everything is working properly, the level in the tank should begin to increase and stabilize at the point determined by the residence time constant and the outputs of the controllers. Check to make sure that the agitator comes on at its pre-set formula level.

LAB #3

WRITE A CL BLOCK

OBJECTIVE

Using the given Blend Tank Simulator Design, code the CL Block and type it into a Text Editor file.

PROCEDURES

Option A

Using the Blend Tank Simulator design specification, code the generic CL Block and the CDS package defined there into one CL package. You will have to create a parameter list for the Flag point entity array. Use the Text Editor to type in the source file. Save the CL package under the name, SIMLVLxx, where xx is your partition number.

Option B

Copy the generic CL source file SIMLVL00 from the instructor volume to your volume. Use the Text Editor to combine the CL Block and the CDS (SIMLVCDS) into a single CL package. Rename the File and the Block SIMLVLxx, where xx is your partition number. Inspect the file in the Text Editor to see if any changes are needed for your point data.

Print a copy of the resulting source file and show it to your instructor. Be prepared to discuss the parameter types and attributes of the custom parameters.

LAB #4

COMPILE AND LINK A CDS PACKAGE AND A CL BLOCK

OBJECTIVE

Compile and link the CL Block to the Custom point in the Blend Tank Simulator.

PROCEDURES

1. Use the MODIFY PATHNAMES display to make sure your partition volume is configured as the CL source/object pathname, and that CDSG is the CDS source/object pathname.

2. Compile the CL Block/CDS package file. If there are any compile errors, correct the problem in the source file (looking at the .LE file to identify errors) and re-compile.

3. When the compilation is successful, list files to find the .LS and the AO files on your volume and the .GD on the CDS volume. Print out a copy of the list file.

4. Make your point SIMLVLxx inactive. Reconstitute it in the DEB and replace the CDS package name with the newly-created one. Enter the proper values in the CDS parameters. Re-load the point to the AM.

5. Unlink the old CL Block from your point SIMLVLxx and link the new CL object file to it.

6. Display the point SIMLVLxx to check that the point data was properly entered and that the CL Block is inserted. Make the point active and check to see that the CL block is executing without errors and that the Simulator is running properly. If there are run-time CL errors, use your list file to try and locate the problem. To correct the problem you may need to unlink, re-compile, and re-link the CL Block.

LAB #5

BUILD FORMULA MANAGER POINTS AND CDS PACKAGES

OBJECTIVE

Using the given Formula Manager design, build Custom points and create CDS parameters.

PROCEDURES

1. Using the Formula Manager design specs and the techniques you learned from the previous exercises, create (write and compile) the CDS packages FMFCDSxx and FMSCDSxx described there. Make as many of the parameters build-visible as you can. Be sure to create the parameters in FMFCDSxx and FMSCDSxx in the same order and using the same names as specified in the Formula Manager design.

2. Create the Custom points described in the design. Attach the CDS packages to the appropriate places. Load the points to the AM and check them to make sure they are built properly by calling up their detail displays.

LAB #6

WRITE FORMULA MANAGER PROGRAMS

OBJECTIVE

Using the given Formula Manager Design, code the CL Blocks.

PROCEDURES

Option A

Using the Formula Manager Design, code the CL sort and load Blocks defined there in the space provided below or on additional paper. It will not matter what order the blocks are inserted on point FMFILExx. Include header information and plenty of comments. In the Text Editor, create CL source files for the two Blocks.

Option B

Copy the CL source files FMSORT00 and FMLOAD00 from the instructor volume to your volume. Rename both files to FMSORTxx and FMLOADxx respectively, where xx is your partition number. Edit both files in the Text Editor to reflect your point data.

LAB #7

COMPILE AND LINK FORMULA MANAGER BLOCKS

OBJECTIVE

Compile and link CL Blocks to the file point in the Formula Manager application.

PROCEDURES

1. Compile and link the CL Blocks to your point FMFILExx. Use all the troubleshooting tools and procedures you've learned about to fix any compile errors.

2. Activate the point FMFILExx.

3. Test the File Manager application by calling up the pre-built schematic FMALPHxx and performing the ADD NEW, MODIFY and LOAD FORMULA functions.

LAB #8

CUSTOM NAME REVISION MISMATCH

OBJECTIVE

Cause a custom name revision mismatch error during point building and find the points which reference the mismatched CDS using Find Names.

PROCEDURES

1. Create the CL source file NET>S0xx>DATAxx.CL by using the listing below.

```
PACKAGE
     CUSTOM
          PARAMETER DATA1:NUMBER ARRAY(1..100)
     END CUSTOM

     BLOCK DATAxx(GENERIC; AT GENERAL)
          LOCAL I:NUMBER

     A:   LOOP FOR I IN 1..100
                    SET DATA1(I) = SQRT(I)
          REPEAT A
     END DATAxx
END PACKAGE
```

2. Compile the source file using the -UL and -OCD options if required.

3. Build two custom AM points DATAxxA and DATAxxB. Specify one CL slot and one custom package. Attach the custom package DATAxx to each point. Give the points a NOPERIOD execution time.

4. Link the CL block DATAxx to each point. Then call up the detail display of each point, make it active and process it once to store data to the array DATA1.

5. Re-edit the CL source file NET>S0xx>DATAxx.CL and add the following parameter statement in the custom data segment definition:

```
          PARAMETER DATA2:LOGICAL
```

6. Recompile the source file using the -OCD option. This causes a new revision number to be given to the CDS.

7. Attempt to build an AM custom point DATAxxC with the CDS DATAxx attached to it. When an attempt is made to load the point you should get the revision mismatch error.

8. Use Find Names or DocTool to search for all occurrences of the CDS DATAxx. Remember that Find Names searches checkpoint files and not the AM's memory. Be sure you checkpoint the AM before attempting to use Find Names.

   Find Names or DocTool should produce an output showing the CDS DATAxx attached to the points DATAxxA and DATAxxB.

9. Now rebuild the points DATAxxA and DATAxxB, while preserving the data in the array. Do this by saving each point to either an IDF or exception build file (your choice). Reconstitute and reload each point with no CDS's. Now reload each point using the IDF or exception build file. This restores the points along with the original data using the new revision of the CDS.

   Call up the detail displays of each point and verify that the data is still in the array.

10. Again, attempt to build an AM custom point DATAxxC with the CDS DATAxx attached to it. When an attempt is made to load the point it should load successfully.

LAB #9

EVENT-INITIATED CL REPORTING

OBJECTIVE

Modify the formula download CL block to generate an event initiated CL report when a new formula is down loaded to the controllers.

PROCEDURES

1. Copy the pre-built Free Format Log source file TICKETxx.DS from the instructor volume to your student volume S0xx.

2. Call up the Free Format Log Editor and read in the source file TICKETxx. Using the add text command put a title at the top of the log which will uniquely identify your report.

3. Compile the source file using the compile command.

4. Copy the log's object file, TICKETxx.FO, to the Free Format Log directory on the network. The directory is NET>FFLG>.

5. The instructor should have already added your Free Format Log to the area database. Verify with your instructor that this has been done.

6. Using the Engineer's Reference Manual, Section 30, as a guide, add the appropriately coded SEND statement to the CL source file FMLOADxx.CL. Include the Console option in the send statement. The statement should cause the Free Format Log to be generated whenever a new formula is down loaded.

7. Compile and link the new version of the CL block FMLOADxx to the point FMFILExx.

8. Call up the schematic FMALPHxx, download a new formula and verify the your Free Format Log is printed at your station.

LAB #10

CNAMEREV ERROR AND RECOVERY

OBJECTIVE

Cause a CNameRev error on the Universal Station and recover from it using the Custom Name Save.

PROCEDURES

1. Notify your instructor that you are at LAB 10. When all students are ready, your instructor will force a CNAME REV ERROR and review the causes and recovery procedures.

LAB #11

PREDICT, MONITOR & OPTIMIZE AM PERFORMANCE

OBJECTIVE

Predict the impact of point and CL data on the AM. Monitor actual AM performance.

PROCEDURES

1. In the empty space provided below, use the Engineers Reference Manual, Sec 22 to estimate the impact (memory requirements and processing load) of all the AM points, CDS parameters and CL Blocks on AM performance.

2. Use the PERFMENU displays (AMDETAIL and/or DATACHNG) to find the following information for the whole AM and for your Unit:

	AM	Unit
Number of points loaded		
Memory used for CL		
Memory used for CDS descriptors and Point names		
Total memory being used		
Cycle where MAX use of CVB occurred in current hour		N/A
Cycle where MAX number of prefetches occurred in current hour		N/A
Cycle where MAX number of points processed in current hour		N/A

3. Using the procedures described in the lecture, identify the cycle where the MAX number of points were processed in the current hour (PRMXCYCC) for the scheduled 1 second, 2 second, 5 second and 10 second points. Also identify the points by either using Documentation Tool or the PSDP parameter AMSCHDMP (0).

LAB #11, PREDICT, MONITOR & OPTIMIZE AM PERFORMANCE Continued

4. Pick a point in your application and re-build it, selecting the CYCLE option and choosing a cycle different from the one listed in your schedule map. Re-load and re-activate the point and do the schedule map dump again to see the changes.

5. Call up the $AMCYLD schematic by pressing the SCHEM key from the operator key board and entering the schematic name.

6. Create a graphical AM loading picture for your assigned unit by selecting the select AM target from the schematic, typing the AM node number, and then ENTER.

7. To create the AM schedule dump file, select ANALYZE AM TARGET, and type 1, then ENTER, enter your assigned unit index number, then ENTER. The AM schedule dump file will then be created and analyzed.

8. Determine the most and least loaded cycle for your assigned unit.

9. Print the list of points on the most loaded cycle.

10. Create a report for the most loaded cycle.

11. Using the $AMCYLD display, familiarize yourself with the different targets.

LAB #12

BUILD NEW DYE CONTROLLERS AND USE DYNAMIC INDIRECTION

OBJECTIVE

Create the three new dye controllers FICDY4xx, FICDY5xx and FICDY6xx. Use dynamic indirection to have the simulator use these new dye controllers.

PROCEDURES

1. Copy the exception-build file FMDYES00.EB from the instructor volume to your partition volume under a new filename. Using the Text Editor, modify it to reflect your point names and use it to exception-build and load your three new dye controllers.

2. Add the three new controllers to an operating group and then call up the group from Operator's Personality and make the points active.

3. Callup the detail display of SIMLVLxx and change the entities in RGPOINTS(2), RGPOINTS(3) and RGPOINTS(4) to the three new dye controller tags.

4. Callup the prebuilt schematic BLDTNKxx and verify that the process controls using the new tag IDs.

LAB #13

CODE A BACKGROUND CL BLOCK

OBJECTIVE

Use the Product Group Selector design specification to write a CL Block that will run at the background insertion point of the formula file point built in Day 3. This block will allow the user to read or write CDS parameter data from twelve separate files on the History Module.

PROCEDURES

Option A

Using the Product Group Sector design specification, code the CL Block RWPRODxx. Include header information and plenty of comments. Use the Text Editor to create the source file for the Block.

Option B

Copy the CL source file RWPROD00.CL from the instructor volume to your volume. Rename the file to RWPRODxx.CL, where xx is your partition number. Edit the file in the Text Editor to reflect your point data.

LAB #14

COMPILE & TEST THE PRODUCT GROUP SELECTOR

OBJECTIVE

Compile and link the product group selector Block to the file point in the File Manager application. Test the function from the Formula Manager schematic.

PROCEDURES

1. Compile and link the CL Block you coded in the previous Lab to your point FMFILExx. Use all the troubleshooting tools and procedures you've learned about to fix any compiler or link errors.

2. From the detail display of the point FMFILExx, enter the pathname skeleton "NET>S0xx>PXX.XX" into TSTRING(2) and the CDS name "FMFCDSxx" into TSTRING(3).

3. Call up the Formula Manager schematic, FMALPHxx. Use the "Write Product Group" target to create several product group files on the History Module. Coordinate the use of product group numbers with other groups since all share the same directory. Read back in another product group to verify the formulas on the schematic change.

4. Change the dye controller tags from the FMALPHxx schematic, then perform a write followed by a read operation. This will cause the newly entered tag IDs to be used by the ratio control. After a change, call up the schematic BLDTNKxx to verify that the process is controlling with the new dye flow controllers.

LAB #15

DETERMINE CPU USAGE BY A CL BLOCK

OBJECTIVE

Given a CL/AM foreground program, add additional code to the program to compute CPU usage time in milliseconds. Use CL/AM Math Library (AMCL02) subroutines.

PROCEDURES

1. Build and load an AM Regulatory point, CPUTIMxx (where xx is your partition number) with the following parameter specifications:

```
UNIT        = AA            CLSLOTS         = 1
PVALGID     = CL            NOPKG           = 2
CTLALGID    = NULL          PKGNAME(1)      = CPUTIME
PERIOD      = NOPERIOD      PKGNAME(2)      = SPQCRAMP
PVEUHI      = 100
PVEULO      = 0
```

   Custom Data Segments CPUTIME and SPQCRAMP already exist on the system.

2. Copy file UNIFORM.CL from the Instructors directory to your student directory.and rename the file CPUTIMxx where xx is your student directory number.

3. Edit your new file CPUTIMxx.CL to change the Block name and End block name to CPUTIMxx. Add required AMCL02 subroutines to calculate the CL block's CPU usage time. Use custom parameter CPUTIME to store the calculated CPU time. Use point CPUTIMxx as your bound data point.

4. Compile your source file CPUTIMxx and link it to data point CPUTIMxx.

5. Call up the prebuilt schematic " CPUTIMxx". PPS your data point , CPUTIMxx, and verify that your program calculated its CPU time usage.

LAB #16

ACCESS CONTINUOUS HISTORY

OBJECTIVE

Given a History Group in which to save Blend Tank Flow Controller and Dye Controller data (History Group built in Lab #1), write and activate a CL/AM program which will access Blend Tank Continuous History.

PROCEDURES

1. Build and load AM Custom Point HISTxx with the following parameters:

```
UNIT            = AA
CLSLOTS         = 1
NOPKG           = 1
PKGNAME(1)      = AVGHISTY  --( existing CDS )
PERIOD          = NOPERIOD
```

2. Chose either Option A or Option B.

   OPTION A:

   Write a CL program that will access Blend Tank Continuous History Averages.

   Specifications: CL Filename:`AVHISTxx.CL`
   Block Name: `AVHISTxx`
   Bound Datapoint: `HISTxx`

   Use existing CDS, AVGHISTY

   Parameter PARM: String
   Parameter BEG_Time: Time
   Parameter END_TIME: Time
   Parameter SIZE
   Parameter STATUS
   Parameter RETRTYPE
   Parameter SCOPE
   Parameter CLSTATUS: Clerrsts
   Parameter Values: Array (1..100)
   Parameter STATUSES: Array (1..100)
   Parameter TIMES: Time Array (1..100)

OPTION B:

Copy file AVHISTXX.CL from Instructor directory NAAM to your directory changing file name to reflect your student directory number. Edit the file to include your Block name and your point name (HISTxx).

3. Compile and link your file to your point. Activate your point Histxx.

4. From the picture editor, read schematic HISTAVxx, where xx is your student directory, from the directory NAAM. Compile the schematic and copy the object file to the PICT directory in the NETwork.

5. Call up schematic "HISTAVxx" to operate history retrieval. Make sure the scope of search is equal to **1**. Set up all necessary parameters, then select the target to initiate history collection and verify that the data was collected from the History Module.