Previse Inc.

Users Manual

Bailey DCS Simulator

Batch Addendum

March 2017

Prepared by:

Previse Inc.®
Managing Critical Microsoft® Technology

Printed in Canada

This document is subject to continuous improvement, and as such is subject to change without notice. Feedback or inquiries regarding this document are welcome. Contact Previse Inc. via www.previse.com.

Table of Contents

Batch Addendum

1 Introduction

This manual is an addendum to the Previse Bailey DCS Users Manual, and describes the Bailey DCS Simulator emulated support for ABB Batch 90 and UDF functions.

Key characteristics of Previse emulated support for Batch 90 and UDF are:

- Execute Batch 90 and UDF programs in ASCII text LST format
- Compile in ABB Batch Manager; execute in Bailey DCS Simulator
- Execute Batch 90 and UDF programs ***as-is***, without changing them
- Execute multiple Batch 90 or UDF programs simultaneously.
- Supports multiple phases within programs, running simultaneously.
- Supports multiple recipe
- Provides Debugger to monitor Batch and UDF execution.

Operator Training Simulator Applications

The Bailey DCS Simulator already provides the ability to execute your DCS CFG logic within a PC based virtual machine as a core Operator Training Simulator component. Now you can also support all your Batch and UDF programs within the OTS as well.

EWS Companion

Together with your EWS (either WinTools or Composer), the Bailey DCS Simulator provides the means to execute your control logic, including your Batch and UDF programs, before you must install into your DCS. All without the cost of putting all the DCS hardware in place

2 Functions and Features

Configuring for Batch and UDF

Configure the DCS structure via an INI file. This INI file defines what Loop : PCU : Module structure the simulated Bailey DCS will have. This INI file defines what CIU devices and MFP/MFC controllers are simulated. Refer to the Appendix for further details of this INI file.

The USB License Key

The USB license key has an option to enable Batch functions (including UDF). To see if your license includes Batch functions, review the Bailey DCS Simulator startup messages at the BaileySim Client Message Tab.

Batch Version

The reference version of Batch 90 for the Batch and UDF functions within the Bailey DCS Simulator is defined by reference [2] and [3] below.

References

[1] Batch 90 and User Defined Function (UDF), Programming Language Reference, (Software Release 4.0), ABB Manual WBPEEUI330252A0.

[2] Batch Data Manager, for Composer (Version 4.2), released November 2004, ABB Document WBPEEUI330253B3.

[3] Batch 90 and UDF Programming Language, for Composer (Version 4.2), released November 2004, ABB document WBPEEUI330254B3.

3 Install, Configure and Operate

Installation

The installation procedure is described in the Bailey DCS Simulator Users Manual. There is no difference for installing a version with or without Batch support.

Before Attempting to Run Batch or UDF

Before attempting to run any Batch application, you must first ensure that:

- The Bailey DCS Simulator is correctly installed
- The license key for the Bailey DCS Simulator is correctly installed and configured
- The default.INI file for the Bailey DCS Simulator is correctly configured to start the simulator, and load all of your CFG files, without any Batch functions.
- There are no installation or configuration issues clearly evident at the Message Tab of the BaileySim client.
- You can start, stop and restart the simulator via the services control, and the simulator starts correctly, with all CFG files loaded, each time.

To do all of these, you must refer to the Users Manual – Bailey DCS Simulator. This Batch addendum does not describe how to perform these steps, but they must be complete before proceeding further to attempt to run any Batch functions.

NOTE: *Ensure that Bailey DCS Simulator runs with all CFG files BEFORE attempting to configure for Batch operations. The Batch functions will not work unless the forgoing steps are completed first.*

Gather the Files Needed for Batch90/UDF Emulation

You must now obtain the files you need for the Previse Bailey DCS Simulator Batch90 emulation. You must follow these instructions to obtain these files from ABB Batch data Manager, as all of the required files must be exported from the ABB Batch Data Manager project:

The files you need are:

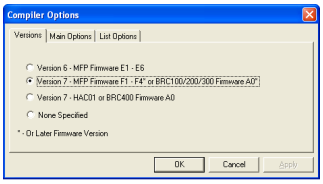
- Unit file (in **.LST** format)
- Program Files (in **.LST** format)
- Recipe Files (in **.TXT** format)

To obtain the Unit Files and Program Files

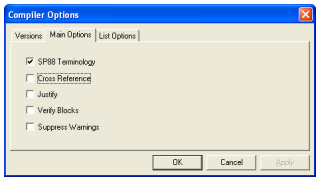
Both the UNIT and PROGRAM files must be obtained in LST file format. LST files are generated as a result of compiling, via ABB Batch Data Manager, the Batch 90 unit(UNT) and program(B90) files.

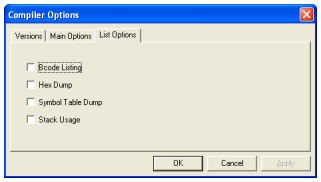
NOTE: *The Bailey DCS Simulator works with the LST files that are generated as a result of program compilation via ABB M\Batch Data Manager. it is required that (a) you must compile your project, and (b) the compilation must be ERROR FREE, before attempting to run the Batch 90 emulation within the Bailey DCS Simulator.*

Obtain the Program Files and the Unit Files



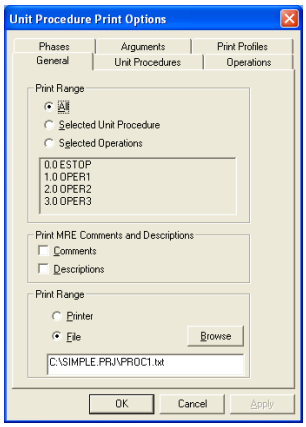
To ensure best emulation results, configure Batch Data Manager according to the following instructions:





1. Run Batch Data Manager and open your Batch Data Manager project.
2. Right click to select the Class in the **Class Library** window, and select **Edit..** from the right mouse menu
3. On the edit window select the **Class/Settings** submenu
4. Change the settings as follows:
   - **Versions** – select any option
   - **Main Options** – Select SP88 terminology, and select no other option.
   - **List Options** – do not select anything.
5. Once the above settings have been made, then compile the unit(<unit>.UNT) file and the program (<program>.B90) file. This compilation generates two files:
   - <unit>.LST
   - <program>.LST
6. These two files must be reviewed to see what ERRORS and WARNINGS are present in the compilation listing. The files MUST have no ERRORS and SHOULD have no WARNINGS (i.e. Batch90 emulation will not work properly if there are unresolved compilation ERRORS, but may work if there are only WARNINGS).
7. Capture these two files for use in the Batch90 emulation.
8. When you are finished you should have exactly 1 <unit>.LST file and 1 <program>.LST for each BSEQ (FC148) in your system.

To Obtain the Recipe Files



To obtain the recipe files you must:

1. Run Batch Data Manager
2. Open the Batch Data Manager project
3. Go to the Unit Procedure / Master Recipe window
4. Select a recipe, press right mouse button, select Edit
5. Select Window / Unit Procedure Editor submenu
6. Select File / Print submenu
7. Print all of the recipes you want to files with .TXT file extension.
8. When you are finished you should have all the recipe that you expect to execute via operator recipe selection.

Copy these files to Batch 90 Emulation Directory

After exporting the unit, program and recipe files per the above instructions, then you should copy each of these files into whatever directory you wish to use for the Batch90 emulation within the Bailey DCS Simulator.

Previse recommends that this directory should be on the same computer that will execute the Bailey DCS Simulation, but not within the **..\Program Files\BaileyDCSSimulator\..** path.

Batch90.INI *Configuration File*

The ***Batch90.INI*** file must be located at the ***..\Program Files\Bailey DCS Simulator\..*** path. You must configure this ***Batch90.INI*** file to list all the Batch90 files required to run your Batch90 programs and recipe within the Bailey DCS Simulator.

Refer to the sample shown here, and create the Batch90 INI files as follows:

```
* program 123 definition
[PROGRAM]
ID=123
FILE=C:\sample\PROGRAM.LST
UNIT=C:\sample\UNIT1.LST
* recipe 101 definition
[RECIPE]
ID=101
FILE=C:\sample\RECIPE1.TXT
UNIT=C:\sample\UNIT1.LST
* unit recipe 103 definition
[UNIT]
UNIT=C:\sample\UNIT103.LST
* program 103 definition
[PROGRAM]
ID=103
FILE=C:\sample\PR103.LST
UNIT=C:\sample\UNIT103.LST
```

1. Start with the **[PROGRAM]** sections of the Batch90.INI file. For EACH BSEQ Block (FC148) in your system, include one [PROGRAM] Section as follows:

   [PROGRAM] – One [PROGRAM] section for each BSEQ in your system

   ID=nnn - this is the program ID for this BSEQ, as defined at FC148 Spec S9.

   FILE=<program_path_and_filename>.LST(.B90) – There should be exactly 1 <program.LST for each BSEQ block.

UNIT=<unit_path_and_filename>.LST(UNT) – For [PROGRAM] section unit file needed to identify target module for current program. It may be any unit file existed for target module or may be created manually. The content of the manually created unit file:

#TITLE <block title, displayed in batch debugger in the list of BSEQ blocks>

#DESCRIPTOR <description>

#LOCATION LOOP=<loop number>, PCU=<pcu number>, MODULE=<module number>, BSEQ=0

Example, file UNITFORPROGRAMX.UNT:

#TITLE " UNITFORPROGRAMX "

#DESCRIPTOR " UNITFORPROGRAMX "

#LOCATION LOOP=1, PCU=1, MODULE=2, BSEQ=0

2. Use ***** as comment lines to separate sections and make the file easier to read.
3. List all of the recipes files in the **[RECIPE]** sections of the Batch90.INI file. For each recipe defined for your system, include one **[RECIPE]** section as follows:

   [RECIPE] – One [RECIPE] section for each recipe defined for your system

   ID=nnn - this is the one of the recipe ID for this BSEQ defined in the UNIT file, as defined at FC148 Spec S1.

   FILE=<recipe_path_and_filename>.TXT – This is the recipe file.

   UNIT=<unit_path_and_filename>.LST(.UNT) – There should be exactly 1 unit.LST for each BSEQ block. In the case of the recipe, which does not related to UNIT definition (in the recipe text file there is the line "Unit Relative: NO"), the unit file has to be created manually. The content of the manually created unit file:

   #TITLE <block title, displayed in batch debugger in the list of BSEQ blocks>

   #DESCRIPTOR <description>

   #LOCATION LOOP=<loop number>, PCU=<pcu number>, MODULE=<module number>, BSEQ=<FC148 block number>

   Example, file UNIT010102.UNT:

   #TITLE "BLOCK222"

   #DESCRIPTOR "UNIT BLOCK222"

   #LOCATION LOOP=1, PCU=1, MODULE=2, BSEQ=222

4. List all of the unit files in the **[UNIT]** section of the Batch90.INI file for all "UNIT recipes". "UNIT recipe" is the used without recipe itself(recipe ID define by output of block defined by S1 equal program ID(S9)). The program in this case consisted of 2 STEP subroutines (STEP SUBR), first used as E-STOP, the second for normal execution.

   UNIT=<unit_path_and_filename>.LST(.UNT)

   All programs for UDF blocks are defined in the **[PROGRAM]** sections of the Batch90.INI file, as described earlier in **[PROGRAM]** section.

5. It could be any number of **[PROGRAM]**, **[RECIPE]** and **[UNIT]** sections in the Batch90.ini file. It is recommended to group it by modules and blocks.

Summary of Batch90 Files Required

The following table lists all of the files that are required to successfully run the Previse Batch90 emulation within the Bailey DCS Simulator

File Types for Batch		
File Name	**File Source**	**Description**
Configuration Files		
Default.INI	Manually created	This file is to configure the controller modules, and the CIU devices, into the Bailey DCS Simulator. This file also supports several other configuration options for the Bailey DCS Simulator. Refer to the Users Manual – Bailey DCS Simulator
Batch90.INI	Manually created	Must be configured to list and identify each LST (unit and Program) and TXT (Recipe) file you wish loaded into the Bailey DCS Simulator.
Program Files (LST)		
xxx.LST(.B90)	ABB Batch Debugger	This is the program file <program.LST> produced by the ABB Batch Debugger when the <unit>.UNT file and the <program>.B90 file are compiled. This file must show that the compilation was ERROR free. You must have exactly 1 of these for each BSEQ (FC148) The <program>.B90 file format may be used, but has to be compiled in order to check for errors.
Unit File (LST)		
xxx.LST(.UNT)	ABB Batch Debugger	This is the unit file <unit.LST> produced by the ABB Batch Debugger when the <unit>.UNT file and the <program>.B90 file are compiled. This file must show that the compilation was ERROR free. You must have exactly 1 of these for each BSEQ (FC148) The <unit>.UNT file format may be used, but has to be compiled in order to check for errors.
Recipe Files		
xx.TXT	ABB Batch Debugger	This is the recipe file exported by the ABB Batch Debugger. You should have one of these files for each recipe that can be executed.

Now You should be Ready to Run Batch90 and UDF Emulation

All the preparation is now completed, and you should be ready to run the Batch90 emulation.

Remember, you must FIRST have run the Bailey DCS Simulator without any of the Batch files, to ensure that all CFG files were loaded correctly before attempting to run Batch90 emulation.

Turn On Batch90/UDF Emulation

As a final step, you must turn ON the Batch90 emulation functions, by setting the Batch90 switch in the ***OPTIONS*** line at the ***default.INI*** file. Refer to the Bailey DCS Simulator Users Manual (Appendix B) for this.

Once this is done, you simply stop the Bailey DCS Simulator execution (if it's already running) and stat the simulator again.

Start The Simulator

The Bailey DCS Simulator Users manual provided startup instructions, which are no different with Batch90 emulation.

Verify that Batch90 and UDF Files Loaded

Verify that the Batch90 files loaded via:

- Review the BaileySim Client Message Tab to see if any Batch90 error messages are issued to suggest that any file failed to load, or if any file was missing.
- Review the Files Tab of the Debugger client to confirm that all files that you expected to be loaded were in fact loaded.

Use Debugger

Use the Debugger to:

- Connect to the Bailey DCS Simulator
- Select and view any BSEQ (FC148) configured into the system
- Observe Batch and recipe operation
- Monitor Operation and Phase status

Shutdown

The Bailey DCS Simulator Users manual provided shutdown instructions, which are no different with Batch90 emulation.

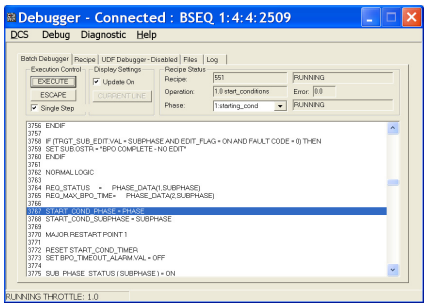
4 The Debugger Client

The Debugger Client provides a graphical interface to the Bailey DCS Simulator.

To run the Debugger application:

1. First verify that the Bailey DCS Simulator is already started and running as a service,
2. Start the Debugger from the Start menu.
3. Multiple Debugger applications can be executed at the same time.

Overview

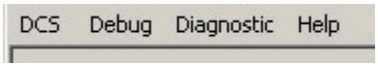


The Debugger provides a menu of selection choices, a status bar along the bottom, and a set of tabs to select various views.

Menu

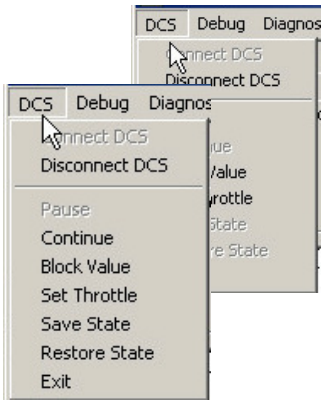
The Debugger has the following menu options:

- **DCS** – Provides operations that affect the entire simulated DCS
- **Debug** – To focus on the specific Batch programs to be debugged, and the specific blocks containing those programs.
- **Diagnostics** – To review the syntax of files to see if all syntax are supported within Bailey DCS Simulator Batch90 emulation (refer to Appendix C for unsupported syntax).



- **Help** – About the Debugger. There is no online help here. This manual, that you are reading now, is the best source of assistance for the Debugger.

DCS Menu Option



The DCS menu option offers several additional options, as follows:

Connect DCS – Connect to the Bailey DCS Simulator, which must already be running. Refer to the Bailey DCS Simulator Users Manual for instructions to start and stop the Bailey DCS Simulator.

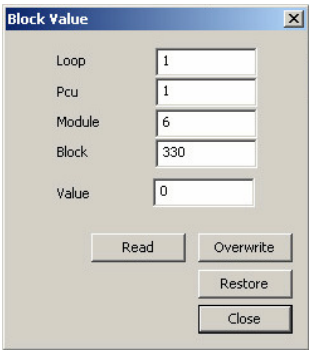
Disconnect DCS – Disconnect from the Bailey DCS Simulator

Pause – Pause operation of the Bailey DCS Simulator. This menu option has the same effect as the Pause button on the DCS Tab of the BaileySim client.

Continue – Continue or Resume operation of the Bailey DCS Simulator. This menu option has the same effect as the Continue button on the DCS Tab of the BaileySim client.

Block Value – Supports changing the value at any address, as with the FORCE function at the BaileySim Client. Controls are as follows:

- **Loop** – select the loop address
- **PCU** – select the PCU address
- **Module** – Select the module address
- **Block** – Select the block address to be changed. This does not need to be the base block address for the function code
- **Value** – the value at the block address (if read) or the value to be written to the address.
- **Read** – read the current value at the selected address



- **Overwrite** – overwrite the value at the selected address
- **Restore** – Restore the values from before this operation
- **Close** – Close the Block Value Dialogue

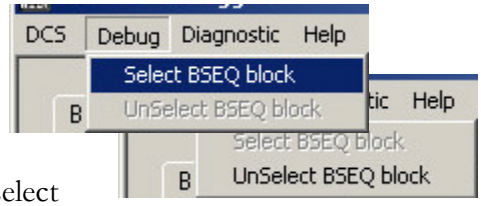
Set Throttle – Set the DCS throttle, as a multiple of 1 X real time. This has the same function as the throttle function at the BaileySim client.

Save State – Save process state, for the entire simulated DCS, as a named state file in the default DCS state file directory. This has the same function as the save state function at the BaileySim client. Refer to the Bailey DCS Simulator manual for instructions to set the default DCS state directory.

Restore State – Restore Process State from the previously saved DCS state files. This has the same function as the restore state function at the BaileySim client..

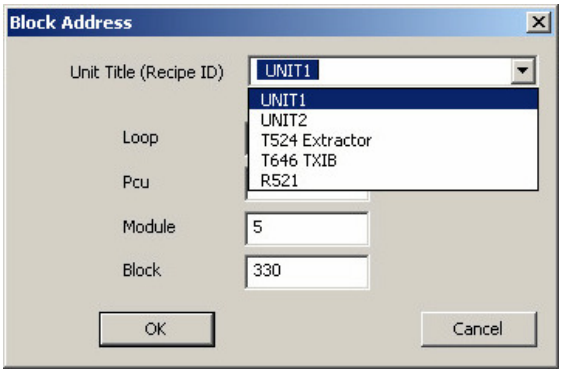
Exit – Exit the Debugger.

Debug Menu Option



The debug menu option allows selection of the particular BSEQ (FC148) to be debugged. There are two menu options:

Select BSEQ Block – This allows you to select which specific BSEQ that you wish to examine. Select the BSEQ from the drop down list on the Select Block dialogue. The L|P|M|B address of the selected block will appear. Once selected via pressing the OK button, the block/BSEQ tab will display the program for the selected BSEQ.



Unselect BSEQ Block - Unselect the BSEQ if you wish to look at another.

Diagnostics Menu Option

The diagnostics menu option supports review of the syntax of any Batch90 LST file to determine which syntax elements are not supported, and displays the result in the LOG tab.

Status Bar

The status bar at the bottom of Debugger shows the following status information:

- Simulator Status – RUNNING or PAUSED
- Throttle - Current Throttle setting

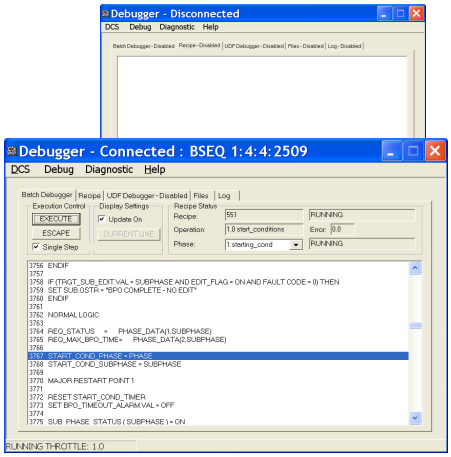
Refer to Bailey DCS Simulator Users Manual for more information about these.

Tabs

The Debugger has five tabs, each providing a different view:

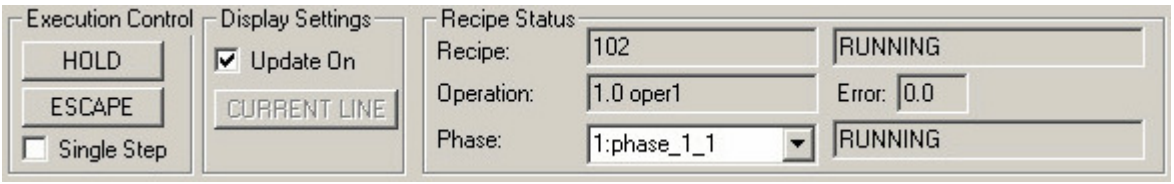
- **Batch Debugger** – Once a BSEQ block (i.e. FC148) is selected, this tab shows the program in the selected BSEQ block, with the L:P:M:B address of the BSEQ showing at the title bar.
- **Recipe** – This tab shows the selected recipe.
- **Files** – This tab shows the files currently loaded into the Bailey DCS Simulator in support of the Batch 90 emulation functions. Each of these files is produced by the ABB Batch Manager software.
- **UDF Debugger** – Once a UDF block (i.e. FC191 or FC192) is selected, this tab shows the program in the selected UDF block, with the L:P:M:B address of the UDF block showing at the title bar.
- **Log** – This tab shows the results of the ***Diagnostics | Check Syntax*** menu action.

Batch Debugger



At startup, the Batch Debugger Tab contains a blank panel, without any program showing. However, once a BSEQ is selected (via menu) the Tab will show the program for the selected BSEQ block.

Once program execution begins, the current line of the Batch program which is executing will be shown in blue.



The controls on the Batch Debugger Tab are grouped into three areas:

- Execution Control
- Display Settings
- Recipe Status

Execution Control Grouping

- **EXECUTE/HOLD Button** – Execute batch program or hold at current line
- **ESCAPE Button** – Escape from FOR, WAIT UNTIL, WAIT FOR, WAIT WHILE or REPEAT UNTIL loop, and move forward in program.
- **Single Step Check box** – Execute a single line at a time.

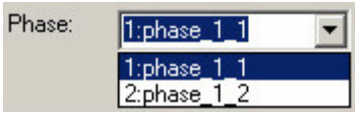
Display Settings Grouping

- **Screen Update On checkbox** – If the screen update is on then the current line being executed will show but it will become difficult to scroll to other parts of the program.
- **Current Line** – When screen update is OFF, this button brings the current line being executed into view.

Recipe Status

- **Recipe:** This is the current recipe selected for this BSEQ (at specification S1 of the FC148).
- **Operation:** - Shows the current Operation name, as named within the Batch 90 program.
- **Operation Status (Upper Right)** - The status of the current operation. This matches closely with the status as would be shown in the native ABB Batch Debugger, as illustrated in the state diagram in Appendix B of ABB manual entitled ***Batch 90 and UDF Programming Language, Composer (Version 4.2)***. For reference, the state will be one of:
  - Batch Complete
  - Normal Logic (Running)
  - Hold Logic
  - Fault Logic
  - Held
  - Restart Logic
  - Emergency Stop

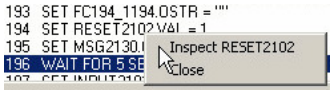
  Refer to the ABB manual for interpretation of these states.
- **Phase:** - Provides the means to select the phase you wish to view, from among the Phases defined for your program.

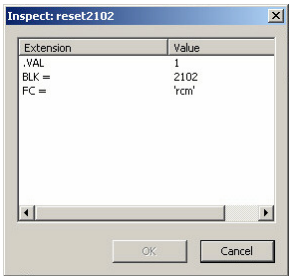


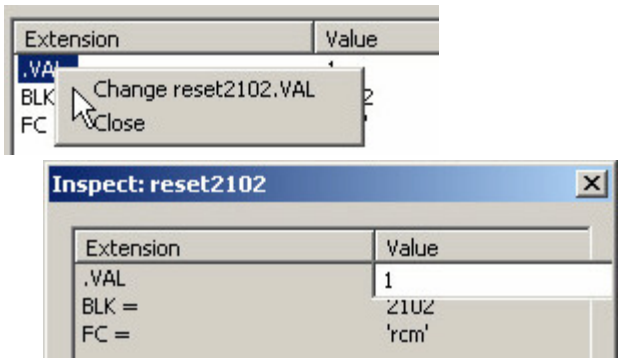
- **Phase Status (Lower right) -** The status of the current selected phase is the same as the status for the Operation (OP above), but as applied to the current selected Phase.
- **Error -** The error, if any, at block output N+7 of the selected BSEQ (FC148) block. Refer to appendix D for an explanation of these errors, which may deviate from native ABB Batch function.

Inspect Variable Function

If you pass the mouse over a variable in the program listing that's shown in the BSEQ tab, you may right click on any variable to inspect the values associated with that variable.

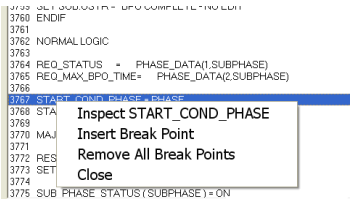






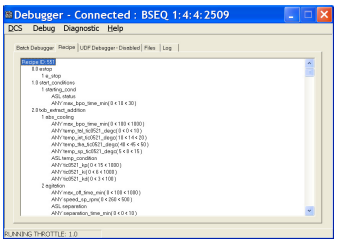
Set and Clear Breakpoint

Right click on any line to bring up the menu to set or clear Breakpoint. Once a breakpoint is set, Batch or UDF will pause execution at the breakpoint so that you can examine status and values at that execution state.



Recipe Tab

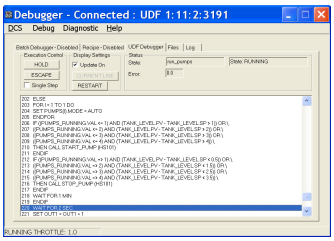
The recipe tab shows the current recipe, as selected at FC148 Specification S1 for the BSEQ displayed at the BSEQ Tab.



UDF Debugger Tab

At startup, the UDF Debugger Tab contains a blank panel, without any program showing. However, once a UDF block is selected (via menu) the Tab will show the program for the selected UDF block.

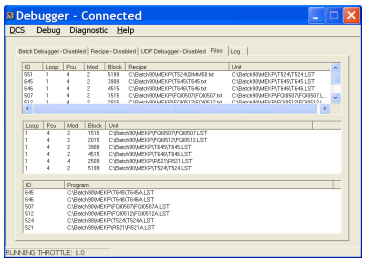
In most respects this tab works the same as the Batch Debugger Tab



Files Tab

The Files Tab shows:

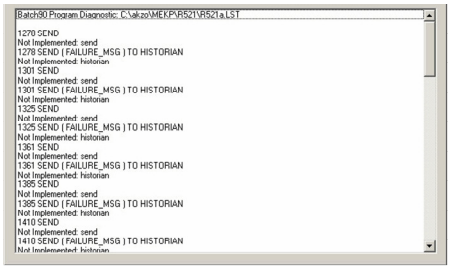
- **Top** – List of <recipe>.TXT files that are loaded, and the recipe ID (per FC148 specification S1) and BSEQ block (FC148) address association for each loaded recipe.



- **Middle** – List of <unit>.LST files loaded, and the BSEQ address association for each Unit.
- **Bottom** – List of <program>.LST files that are loaded, together with Program ID number (per spec S9 of FC148).

Log Tab

The Files Tab shows the results of the Diagnostics syntax check. Refer to Appendix C for a summary of unsupported syntax. Contact Previse if you need support for any feature not yet supported.



Appendices

Appendix A – Batch Event Messages

The following table lists additional messages that may be sent to the BaileySim Client Message tab page in support of Batch functions. Refer to Appendix C of the Bailey DCS Simulator Users manual for other event messages.

All messages arising from the emulated emulation of Batch90 programs within any BSEQ block are prefixed with the string Batch90:

e.g. ***Batch90:*** *Batch90.ini file has not been found*

Event Messages in Support of Batch Functions	
Message	**Interpretation**
INI File Issues	
Batch90.ini has been loaded Batch90.ini file has not been found	Ensure Batch90.INI file is present
File Related Issues	
Unit file <filename> has not been found Program file <filename> has not been found Recipe file <filename> has not been found Recipe <filename> not loaded(Unit is not loaded)	Ensure correct files are at the path as specified in batch90>INI
Batch Program Syntax Issues	
Error/Warning :Unit Load/:Program Load/:Recipe Load/in BSEQ (<L:P:M:B>) RecipeID:<id> ProgramID:<id> Line:<id> <text>	Syntax issue at line as noted. Verify syntax is correct, and contact Previse is issue persists
Error/Warning :Unit Load/:Program Load/:Recipe Load/ in BSEQ(<L:P:M:B>) RecipeID: <id> ProgramID: <id> <text>	Syntax issue at location as noted. Verify syntax is correct, and contact Previse is issue persists

Appendix B – FC148 Errors

This appendix provides a list of Errors support at output N+7 of each FC148 within the Bailey DCS Simulator. This list also annotates which standard errors within a real FC148 are not emulated.

FC148 Errors at Output N+1	
Error	**Meaning**
Errors in range +1, +2, +3 etc	
+1, +2 +3 ….	**User Defined Error** - This number (any positive number) is assigned by the user in the batch language program and is used to indicate what type of fault has occurred. There is no limit to the number of fault codes the user may assign.
Errors in range -1, 12, -3 to -32 (These errors are emulated from BSEQ Errors as defined by ABB)	
-1	**Hold command** - The batch sequence is in hold, through either the BSEQ function block or a command in the batch language. Going to hold suspends normal logic and starts execution of hold logic.
-3 (*)	(*) This ABB error is not supported in Bailey DCS Simulator
-4	Program specified at FC148 Spec S9 does not exist. Verify that you have gathered all the PROGRAM and UNIT LST files (see Appendix A) and that the Batch90.INI file is correctly coded.
-5 (*)	(*) This ABB error is not supported in Bailey DCS Simulator
-8	**Recipe refers to undefined phase subroutine** - The unit recipe contains a phase subroutine name that is not contained within the batch program. This situation can happen when a batch program is edited so that a phase subroutine is removed, but the corresponding recipes are not changed. To correct, add the undefined phase subroutine or remove the called (undefined) phase subroutine from the unit recipe.
-10	Recipe specified at FC148 Input defined by Spec S1 does not exist. Verify that you have gathered all the RECIPE TXT files (see Appendix A) and that the Batch90.INI file is correctly coded and that the Recipe ID has been correctly entered..
-12	**Illegal parameter type** - There is an argument data type conflict between the unit recipe and the batch program. To correct, recompile the batch program and the recipe. Then, download both the recompiled batch program and unit recipe.
-13	**ESTOP/Aborting from block input** - The emergency input to the BSEQ function block (specification S5) is on. This drives the batch program unconditionally to operation 0 of the current unit recipe. To correct, find out why the emergency shut down input is being set to on and correct it.
-15	**Invalid operation number** - The starting of a batch sequence at an operation number not defined within the unit recipe being run was attempted. Create a unit recipe with an operation number that matches the one to be executed or change the operation number.
-16	**Bad function block reference** - In the batch data declaration sections of the batch language, the program is trying to reference a function block that does not exist or one whose type does not match the function code type in the declaration. The batch debugger will provide the function block number within the batch data / that is making the reference. To correct, change the function block number to a valid one, erase the reference from the program, or correct the type to match the function block in the controller.
-17	**Array error**. - An array subscript is out of bounds. Normal logic is suspended and execution of fault logic begins. It is possible to inspect the value of the fault code to detect when this fault has occurred
-18 (*)	(*) This ABB error is not supported in bailey DCS Simulator
-19	**Recipe requires too many parallel phases** - The execution of a recipe that contains more parallel phase subroutines than are allowed in the target program was attempted. To correct, edit and recompile the unit recipe to contain less parallel phase subroutines or edit the #MAXPARALLEL statement.

FC148 Errors at Output N+1	
Error	**Meaning**
-20	**Invalid number of parameters in phase data** - The unit recipe contains the wrong number of recipe parameters as compared to the target program. To correct, fix the recipe or the program (normally the recipe).
-25	**Bad block reference in phase data** - The unit recipe used contains a reference to an incorrect or nonexistent block. Correct the block number in the unit recipe.
-26	**Bad data reference in recipe** - A data entry in the unit recipe does not match the program. This most commonly happens when a unit recipe argument value was selected from a selection list and the program was changed to no longer include that selection. Resolve any discrepancies and recompile the unit recipe with the batch program.
-27	**Bad block reference in Unit Data** - A function block declaration in the unit data file does not match the controller configuration (either the function block address or the function code type are in error). Resolve any discrepancies and recompile the unit data file against the batch program.
-28	**Bad CSEQ reference in Unit Data** - A CSEQ reference in the unit data file does not match the program. Resolve any discrepancies and recompile the unit data file against the batch program.
-29	**Unit Data does not match Batch90 Program** - The unit data file (Unit.LST) does not match the batch program (Program.LST) structurally (the number or type declarations do not match). Resolve any discrepancies and recompile the unit data file against the batch program. It's possible you provided the wrong files.
-30	Unit file that associated with the Recipe ID (per FC148 Spec S1) does not exist. Verify that you have gathered all the UNIT LST files (see Appendix A) and that the Batch90.INI file is correctly coded.
-31	**ID type mismatch** - The recipe ID type connected to the BSEQ function block is not the same as the program file type selected by BSEQ specification S9. This error is caused by the BSEQ input specification S1 connected to a User Defined Data Export (FC194) and the program referenced by specification S9 being Numeric or specification S1 connected to a read and the program referenced is #alpha.
-32	**String subscript error** - The string position specified in the program is negative or larger than the maximum size of the string. This error occurs during program execution and transfers the program to fault logic.
Bailey DCS Simulator specific errors have error number starting at -101	
-101	**BSEQ syntax error** - Refer to event log message for address of error.
-102	**Unit recipe definition error** – This error indicates that there is a mismatch between the UNIT file and the RECIPE file. You should ensure that the correct files have been produced (per Appendix A) and copied for use within the Bailey DCS Simulator. You also need to confirm that the compilation was ERROR free.
-103	**Program is too long for execution** - This is intended to identify infinite loops within Batch 90 programs that will never exit.

.

Appendix C – Unsupported Syntax

The following table identifies Batch90 syntax elements that are not currently supported. If you need support added for these, please contact Previse.

Unsupported Syntax	
Syntax	**Comment**
ALPHA	
AUX	
BSEQ	Will be implemented upon end user need and request.
BUFFER	
CASESENSE	There is no need to support this Compiler Directive in Bailey DCS Simulator.
COMMON	
CYCLETIME	Will be implemented upon end user need and request.
DEBUG LEVEL	Debug level is not supported. Simulator defaults to DEBUG LEVEL 4
DEBUG_TOK	
EJECT	There is no need to support this Compiler Directive in Bailey DCS Simulator.
ENDDATA	Will be implemented upon end user need and request.
ENTRY	Will be implemented upon end user need and request.
ESTOP	Will be implemented upon end user need and request.
FGEN	Will be implemented upon end user need and request.
HIHI	
HISTORIAN	Batch Historian functions not supported within Bailey DCS Simulator.
HISTORIAN BUFFER	Batch Historian functions not supported within Bailey DCS Simulator.
HISTORIAN STATUS	Batch Historian functions not supported within Bailey DCS Simulator.
INCLUDE	There is no need to support this Compiler Directive in Bailey DCS Simulator.
INVERT	Will be implemented upon end user need and request.
LOCATION	
LIST	There is no need to support this Compiler Directive in Bailey DCS Simulator.
MAXPARALLEL	Maximum number of parallel subroutines is set to 32 by default.
NOLIST	There is no need to support this Compiler Directive in Bailey DCS Simulator.
NORMALIZE	
RUN	Will be implemented upon end user need and request.
SEND	
SIZEOF	Will be implemented upon end user need and request.
STARTUP	
STATE	
STATUS	
SUPPRESS	
TIME	
TITLE	There is no need to support this Compiler Directive in Bailey DCS Simulator.
TUNED	
UDF	UDF Function Code 193 not supported (remainder of UDF blocks supported)
VERSION	
PHASE NUMBER	Will be implemented upon end user need and request.
RECIPE ENTRY	Will be implemented upon end user need and request.
SUBR DESCRIPTOR	Will be implemented upon end user need and request.
SUBR STATUS	Will be implemented upon end user need and request.