

1. Battery Console

Battery Console Software Documentation

V1.0.0.91

This documents the Battery Console software application used to manage Battery Metric devices.

Battery Console is compatible with the MC, LC, RM and MK series of battery test modules and analyzers.

Users should be running firmware version 9.6 or greater. Check: **Menu/System/Configuration** to verify firmware requirements. Update firmware if required. See **Firmware Updates** section in device user guides for more info.

1.1 Read This First

The Battery Metric analyzers and modules together with the associated software tools create a system that allows the user full control to specify many battery charge, discharge and test parameters. In addition, the system has options to modify the calibration by the user. These capabilities are designed to provide the user with maximum flexibility to manage many different battery types in many different situations. This capability, by its very nature, also creates the possibility for the user to use improper charge, discharge, test or calibration parameters. When used improperly, by accident, inadvertently, by mistake or with lack of proper knowledge and training or if the software files become damaged, the system can fail to properly manage the battery under charge, discharge or test. This can damage batteries and as a result a battery could leak, overheat, explode or catch fire. This could create a dangerous situation involving fires, property damage, personal injury or death.

It is expected that this system will be used with this information in mind. The user understands this situation and agrees to use due diligence when using this system. The user should only charge LiIon and LiPo batteries that have built in protection circuitry to protect against the possibility of applying improper voltage or current to the battery. Most consumer type LiIon batteries do have built in protection circuitry. If you are uncertain if the LiIon or LiPo battery has built in protection circuitry, check with the battery manufacturer. Always charge, discharge and test all batteries according to the battery manufacturer's specifications. Users should verify that proper voltages and currents are being applied with their own voltage and current measurement equipment.

This manual is a guide to using the Battery Console software application. It is not a training manual for understanding proper battery management, charging, discharging and testing. It is expected the user will have this knowledge. The Battery Metric system is a battery management platform to be used by knowledgeable trained staff. It is not a consumer type product. Read and follow the Safety Precautions listed in Safety section.

Users who purchase a Battery Metric module or analyzer receive a license to use the Battery Console software programs. You must read and agree to the license agreement listed below before continuing with the use of the software. Users who receive this software with an analyzer or download from our web site have a license to use it subject to the license agreement listed below:

Under the terms of this license NO warranties of any kind are made with respect to the contents or functionality of this software, nor its fitness for a particular purpose. LaMantia Products Ltd. shall not be liable for errors in the software or documentation, or for any direct, indirect or consequential damages or financial losses arising from the use of this software or the Battery Metric analyzer modules.

Read and follow the safety precautions specified here and recommended by the battery manufacturer. Also

follow the battery manufacturer's recommended charging and discharging procedures. LaMantia Products Ltd. reserves the right to change the specifications and software without notice. Users agree to make their own determination as to the suitability of this analyzer for their particular application.

You accept the terms of this license by continuing with the use of the software. If you do not agree to the terms of this license then return your analyzer to your dealer within 30 days of purchase for refund.

The software program Battery Console for Windows™ and its associated documentation are copyright © 2018 LaMantia Products Ltd. London, Ontario, Canada. All rights are reserved. Unauthorized duplication or distribution is strictly prohibited.

Before attempting to use your Battery Metric device please read all users manuals supplied with the device.

1.2 Safety

Safety Precautions

Rechargeable batteries can store tremendous amounts of energy. This can be a fire hazard or result in bodily injury, property damage and death if mistreated. If at any time you need to stop the analyzer simply click the on-screen *Stop* button. This will halt all charge and discharge current and return the analyzer to an idle state. Observe the following precautions:

- Only use the device to charge secondary rechargeable batteries. Other types of batteries may burst causing personal injury and damage.
- The device should not be used for charging any kind of primary cells, including reusable alkaline batteries. It can be used to discharge primary batteries.
- Always follow the battery manufacturer's recommended charging and discharging requirements.
- Never short-circuit the terminals of a battery.
- Do not leave batteries unattended while charging. This is especially true for LiIon and LiPo battery types. These battery types can explode and cause fires, damage property, cause personal injury and death.
- Consider using an explosion and fire resistance container to house the battery under test if required. This may include but is not limited to charging LiIon and LiPo batteries in instances where the batteries' protection circuitry is suspect or under test.
- Do not defeat any special protection features or circuitry in the battery or battery pack.
- Use your own voltage and measurement equipment to validate current and voltage calibration. Re-calibrate if required.
- Only charge LiIon and LiPo batteries that have built in protection circuitry to protect the battery from improper voltages and currents that may be applied for any reason.
- Always observe correct polarity when connecting batteries.
- Remove batteries from all equipment. Do not connect electronic equipment to a *Battery Metric* device. Failure to do so could damage your equipment and will void the warranty.
- Keep the device in a well ventilated location such as on a table top, open shelf or rack. The device will generate heat during use and as such should not be used in an enclosed area. Do not use it in an enclosed cabinet or tight shelf area. Keep power supplies well spaced and ventilated.
- Fully test all custom routine programs to ensure that they operate as expected in all possible situations

that may arise.

- Disconnect batteries when the device is not in use.
- Only use the device in a dry location.
- Do not connect a battery with a voltage greater than the analyzer's rated voltage.

2 Introduction

The Battery Console software application is used to control and manage the Battery Metric analyzers and test modules. Use Battery Console to:

- Independently manage all channels
- Select test parameters
- Manage database of test profiles
- Load & run standard *Routines*
- Load, preview and run custom program *Routines*
- Operate device to Start, Stop, Restart test sessions
- View Real Time data
- View Saved Data results
- View Real Time chart
- Collect data and results log files
- Print test reports

A *Routine* is a battery test program designed to perform a specific task for a specific battery type and configuration. It can charge, discharge, perform tests, check parameters, calculate battery performance, and provide results. Powerful conditional routing, tests, cycles and other tasks can be accomplished for creating intelligent battery test functions and algorithms.

As the test session is proceeding, results are conveyed to the user via the On Screen console, LED's and Beeper. The On Screen console provides more information than you would normally see on a typical LCD panel and can be customized for each routine.

Once the routine is loaded, the device can also operate on a stand alone basis without requiring the host PC connection. The LED's and Beeper provide status and results information. Build a battery management product to perform specific tasks.

Battery Console contains standard routines that can be used by just selecting the desired battery parameters. Alternatively custom routines can be programmed for maximum flexibility. The Routine can be customized for a unique application. This basically gives you a programmable battery manager with a level of control that is truly impressive. By having one device that can be reprogrammed you can perform many different functions now and in the future.

What can Routines be used for ?

Use the device for a simple charge or discharge session or for more complex functions. Limited only by your imagination, here are some examples to get you thinking:

- Battery charger
- Battery discharger
- Battery load module
- Production commissioning
- Special test equipment
- Quality control
- Go / No Go testing
- Pass, marginal, fail grading
- Protection circuit testing
- Battery fuel gauge calibration

- Battery Conditioning
- Capacity measurement
- Internal Resistance measurement
- Cycling
- Unique battery algorithms
- Charge monitoring
- Voltage monitoring
- Life cycle testing
- Identify under performance
- Warranty validation
- Battery selection
- Acceptance testing
- Lost chargers
- Regulated power supply
- Delivery Preparation
- Over voltage monitor
- Temperature alarm
- Float/trickle charging
- Threshold alarms
- Absorption charging
- Cell monitoring
- Expanded scale volt meter
- Power supply
- Constant power discharging
- Constant resistance discharging
- Conditional routing
- Intelligent battery manager

Resources available on Battery Metric analyzers and modules

Enjoy a wide array of battery management resources. Take advantage of the powerful routing, looping and comparison functionality to meet any charge, discharge or special test function. Program custom routines to have full control over the device's various resources for a special test or application.

- Current regulator
- Voltage regulator
- Constant current charging & discharging
- Constant power discharging
- Constant resistance discharging
- Current measurement
- Voltage measurement
- Temperature measurement
- Flash program memory
- mAH / AH measurement Time measurement
- Many termination options
- 7 programmable counters
- 3 timers
- Programmable LED's
- Programmable Beeper
- Data retention
- Customize Console Display
- Manual Vector
- Programmable routing
- Parameter checking & routing
- IR measurement function
- Custom user messages
- Programmable IF and GoTo statements
- Conditional routing
- Programmable pulsing
- Data logging
- Current Sharing across channels
- 2 programmable LED's
- High Speed Pulsing
- Many termination parameters
- Programmable Results messages
- Programmable user instructions

Use the Battery Console application to write or edit a unique battery test routine. This is great for setting up a special task in a production environment. Once the routine is created and loaded into the device, production staff can easily use it without special training or knowledge. Great for commissioning batteries or packs or other routine quality or production requirements.

On screen data, controls and results are at your fingertips. If it's pure battery functionality that's required then the routines can operate in stand alone mode for fast and easy production tasks.

3 Installing Battery Console

The Battery Metric software application installation files can be found at: www.batterymetric.com

To install Battery Console simply download and run the setup file: [setup.exe](#)

This installation method will require an internet connection.

If an internet connection is unavailable, download the zip file containing all of the installation files. Unzip the files to a directory and then run the setup.exe file from that directory.

The application will automatically check for updates each time it is started. If the automatic updates are cancelled, when prompted, then no further checks will be made for updates. To turn the automatic updates back on, uninstall the application and reinstall with a new setup file.

The software will run on all modern 64 bit Windows operating systems.

4 Screen Layout

The on screen console displays operational information as the test session proceeds and session results when complete. Controls allow you to load routines, move between channels, control the test session and other functions. The Real Time Data area displays information about the session as it is happening. The Saved Data area displays information about completed steps.

The following highlights the different areas of the main screen



Note: The Window title bar at the top displays the description of the Routine currently loaded.

Section	Description
1-Main Menu	Access for control and other options.
2-Info Fields	Model No and Online Status of selected channel Function the device is currently performing. This could be: Charge, Discharge, Pause, IR Test, Stop etc.
3-User Message	Helpful details about what's happening and instructions to the user.
4-Real Time Data	Displays real time information from the session as it is happening.
5-Results Data	Results that the device has saved for reporting purposes. This data is saved in the device and will continuously be available until a new session is started. This saved data is available even after device power off/on and the software is restarted. In this way a device can be used in stand alone mode to perform a battery management function and then report the data at a later time if required.
6-Controls	Controls used to operate the device. The leftmost 3 buttons are for loading standard routines. The function of these 3 buttons change when in Program Setup Mode. The right 3 buttons control the routine once it has been loaded.
7-Chart Area	Includes Voltage, Current and Temperature plots.
8-Status Bar	Shows important system configuration details for reference.
9-Chanel Number	This is the channel currently selected. In order to operate the device you need display the correct channel. Use this channel display to ensure you are on the desired channel.
10-Channel Selector	Use controls to move between channels. Note that you can also change channels simply by placing cursor in Channel Number display, enter desired channel number and press enter.
11-Routine Help	View the help memo that is programmed into the device. This is useful information to operator about how to use the currently loaded routine and what to expect.
12-Compact View	Makes window smaller by hiding the Setup and Chart areas to reduce screen clutter.
13-Quick Setup Mode	Select this tab to enter Quick Setup mode. Select this mode of setup to use default test parameters based on only 3 inputs: Battery Type, No of cells in series and rated capacity. Once these input have been made any of the standard test routines can be loaded in the device. After loading the device then the Start , Pause and Stop controls can be used begin and end sessions.
14-Profile Setup Mode	Select this tab to enter Profile Setup mode. Profile Mode setup allows for full control of all the battery management parameters. This tab is used to manage the database that contains all of the saved profiles created. Users will need to create, modify and edit the profile parameters as required. The software is supplied with a few example profiles but these should be adjusted as required. To use the profile simply keep this tab active and any of the standard test routines can be loaded into the device. After loading the device then the Start , Pause and Stop controls can be used begin and end sessions.
15-Program Setup Mode	Select this tab to enter Program Setup mode. This will enable the Program Menu selections and the 3 left most controls will also change for Program management. Once Program Mode has been activated, they can then be opened, previewed and loaded for use. After loading the device then the Start , Pause and Stop controls can be used begin and end sessions.
16-Device Setup Area	This is where all test parameters are set and viewed. The area changes depending on what setup mode is currently selected
17-Profile Controls	In Profile Mode this area will contain the Create, Copy, Delete controls for managing the Profile database. In other setup mode this area displays helpful information.

4.1 Menu

Use the *Main Menu* to access various controls, functions and set options.

File

Print Results: Use to produce a printed report. This includes the Results Data as well as operational information about the setup parameters.

Print Chart: Print a copy of the chart as it appears on the main screen.

Clear Device: This will erase the routine current loaded into the device. A new routine will have to be loaded the next time a new session begins.

Quit: Exit the Battery Console application.

Routines

The Routine menu contains a list of standard routines that can be used to manage the battery. Selecting any of these routines will begin the process of loading the routine into the Battery Metric device. Note that the routine typically does not begin just by loading it. Usually you will need to begin the session by clicking the Start button if a battery is already connected. Alternatively the routines are setup to be started if a new battery is connected. If a battery is already connected when the routine is loaded then it can be started by removing and reconnecting the battery. By having the session begin with the connection of a battery allows for operation of the device without being controlled by the computer. This creates the opportunity to operate the Battery Metric device in a stand alone situation separate from the computer. Of course you still need to use the Battery Console application to load the routine, but once the routine is loaded it can be used and reused repeatedly by just connecting new batteries. The routine will reside in the Battery Metric device and persist even after the power is turned off. By simply powering up the device it is ready to continue to be used for new Battery Test sessions without having to reload the routine.

Charge: This will load the Charge routine required for the type of battery and other parameters specified in the setup parameters.

Discharge: This will load the Discharge routine required with the discharge parameters specified in the setup parameters.

Charge: This will load the Charge routine required for the type of battery and other parameters specified in the setup parameters.

Cycle: This will load the Cycle routine required for the type of battery and other parameters specified in the setup parameters.

Recover: Some routines will not operate if the battery voltage is out of range. This routine can be used to apply a recover charge to a depleted battery. The duration is specified in the profile.

Load Test: Perform a load test using the discharge current and discharge interval specified in the parameters.

Standard Routines are only available when in Quick Setup or Profile Setup mode. Click the appropriate setup tab to select setup mode.

Program

This menu section is used to manage a custom program routine.

Open: Opens the xml program file so that it can be previewed or loaded into the Battery Metric device.

Preview: Opens a new window that displays a comprehensive display of the all the programming parameters. This display is very helpful for developing, reviewing and understanding how the program will operate. This preview window is for display purposes only and cannot be edited. The editing must still be done with an XML editor program. However this window is a more convenient way to visualize the total program operation. When programming a new routine the best practice is to keep the XML editor open and as you save changes in the XML file, reopen the preview window to review the changes.

Load: This will compile and load the program file into the Battery Metric device. The program will then be routed to step 1 and be ready for operation.

Clear: This simply clears the program workspace.

Force: Use this to force the loaded program to route to a specific step. Select Force then enter the desired step number. This can be used in any way required. It could be part of the how the program is designed or it could just be use to force a new step for development and debugging purposes. Note that a program could be designed with many different functions and the Force feature could be used to instigate a change to the new function. In effect you can have multiple programs or task written into a single xml program file. Switching between functions or task would be as simple as forcing the device to a new step.

New: This will create a new blank xml file template that can then be edited with an xml file editor. XMLNotepad is a recommended xml file editing program. This is a free editing tool from Microsoft. Download here: [XML Notepad 2007](#)

Program controls are only available when in Program setup mode. Click the Program tab to enter Program setup mode.

View

Compact Mode: Used to shrink the user interface by closing the Setup and Chart areas. This can be used to reduce clutter on the desktop while working on other applications.

Graph Enable: Used to hide the chart area when not required.

Graph Voltage: Used to display or hide the yellow Voltage plot line.

Graph Current: Used to display or hide the blue Current plot line.

Graph Temperature: Used to display or hide the red Temperature plot line.

Graph Color: Used to change the background display color of the Chart area. To restore the default Grey color select Cancel.

System

Configuration: Opens a new window that will display the Channel No, Model No, Serial No, Firmware Ver, Hardware Ver and Range of all devices currently online. Consult this information to determine which channel numbers are currently available. When using the system be sure to know which channel numbers are assigned to all devices. When using multiple channels it is a good idea to label each channel with the corresponding channel number to avoid confusion. When updating the firmware check this configuration information to ensure the firmware has been updated successfully.

Results Logging: Use this to enable results logging. When enabled, results from each step, that is configured to save data, will be written to a log file. The location of this log file will be the same as specified in the Data Logging window. The results file is a TXT text file that can be opened and viewed with any text editor program such as Word or Notepad. Results logging is useful when running many steps, such as life cycle testing, where

the results cannot all fit in the Results Data area of the main screen.


Data Logging: Use this to enable data logging and also specify the file location where the file should be created. The data logging file contains a data stream of real time data from the analyzer. This file is of the type csv (comma separated variable) that can be imported and managed with many popular spreadsheet programs such as Excel.

Use mA Range: Use this to toggle data fields to display data in either mA or A.

Use ABS Values: Use this plot discharge current as a negative value. This means discharge will plot below the white X axis and charge will plot above the X axis. It's just for convenience to make the plot area more intuitive to match user preferences.

Show WH results: This is used to display either AH or WH the Data Results section.

4.2 Real Time Data

Real Time Data: 					
Voltage:	4.016	Current:	-1000	Step Time:	00:02:16
Total Time:	00:02:16	Cycle:	1	Step No:	4
mAH:	-37.6	Result:	-17.1 %	Count 2:	0
Bat Temp:	24.3 °C	W:	-4.0	WH:	-0.2
Int Temp:	34.9 °C	Spare:		Spare:	
Count 3:	12				

Real Time Data is measurement data that is collect every time *Battery Console* polls the device. The default poll interval is once/sec. It provides a dashboard for the battery test session as it is proceeding. This display provides an instant picture of the situation and what is happening at any given time. Data includes instantaneous measurements and accumulated data for the step currently operating. Use ↓ button to show more fields. Use ↑ button to hide fields.

The following fields are included in the Real Time Data display:

Data Field	Description
Voltage	Battery Voltage in Volts.
Current	Current in mA or A. Discharge current has a neg sign unless the ABS option is enabled.
Step Time	Time duration of the present step. Format:HHH:MM:SS
Total Time	Total elapsed time of the session. Session starts when the reset step is executed. Format:HHH:MM:SS
Cycle	This is Counter 1 which is typically used as a cycle counter.
Step No	The Step No currently in effect.
mAH	Input or output charge expressed in mAH or AH depending on option used. Discharge uses a neg sign unless the ABS option is enabled.
Result	The actual AH measurement as a percentage of the <i>Rated Capacity</i> .
Count 2	General use counter. It is cleared at the beginning of each new session.
Bat Temp	Battery temperature in °C when using optional temperature measurement cable.
W	Power in Watts being delivered to/from battery.
WH	Energy in Watt-Hours being delivered to/from battery.
Count3	A special counter that only gets reset on power off/on. It does not get cleared with a new

session. It can be used as a session counter for daily/weekly production counts. This counter is incremented at the beginning of each Reset step.

Int Temp Temperature inside device as measured near heat sink.

4.3 Results Data

Results Data:						
#	Function	Time	mAH	Results	Term Type	Message
1	Discharge	00:34:03	-200.1	90.9%	AH	Pass
1	Charge	00:34:07	200.0	90.9%	AH	
1	IR	00:00:06	---	247.1	mΩ	
2	Discharge	00:34:03	-200.0	90.9%	AH	Pass
2	Charge	00:34:07	200.0	90.9%	AH	

Results Data is the results from each step that completes. As a session is proceeding you will see data collected and accumulated in the Real Time data section. Once that step completes the accumulated data and appropriate messages are saved by the device. The Results Data from the completed step are then transferred to the Results Data section for reference. The Results Data also appears on printed reports. The Results Data is retained until a new session begins, at which time the old session data resets.

Results Data is retained by device and is available for viewing even if the Battery Console application is started during or after a test session. This allows the device to be used even without a PC connection. The Battery Console application can be started later if/when viewing the data is required.

When Battery Console is running, and the Results Logging option is enabled, all Results Data also gets written to the Results Log file. The Results Log is a useful permanent record of the session. The maximum number of steps/rows available in the Results Data section is 64. The Results Log file can record any number of step results.

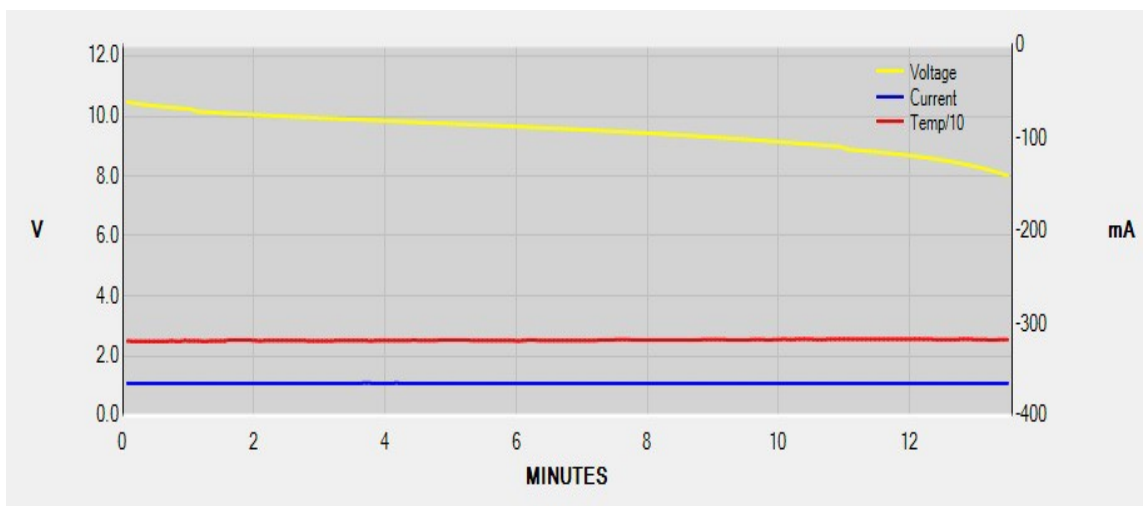
Use ↓ button to show more rows. Use ↑ button to hide rows. Use the scroll bar to view additional rows. Each line of the Results Data represents the completion of one step. It's organized with 1 row for each step completed. The following columns are included:

Column	Description
#	This is Counter1 that is usually used as a cycle or step counter.
Function	The function executed on this step. ex. Charge, Discharge, Pause, IR Test, Stop
Time	Time duration of the step. Format:HHH:MM:SS
mAH/AH/WH	Displays result depending on option selected. See: Menu/System/...
Term Type	Indicates what event generated the step termination.
Message	A custom message used to inform the operator. See <i>Program Messages</i> section for examples.

When preparing custom routines be aware that you must Save the data for any step that is to appear in the Results Data. Do this by setting Save=yes for the desired steps in the xml program file. Also note that Counter 1 is a special counter that gets saved with the Results Data and is used to assign a number(#) to the Result Data row. Since the *Results Data* is saved to memory before counters are incremented. It is necessary to increment

Counter1 before beginning the step that will eventually correspond to Counter 1. Keep this in mind when programming custom routines to ensure that Counter1(#) will match with the correct Results Data row.

4.4 Graph Data



The Graph area will display a plot of Voltage, Current and Temperature as the battery test session proceeds. The chart is updated in real time as new data is received. The plot lines can be enabled/disabled and background color changed. See: Menu/View.

Voltage is read on the left axis. Current is read on the right axis. Temperature divided by 10 is referenced to the left axis. The left axis maximum can be adjusted if required to get a better plot view. Using mouse left click will increase left axis maximum. Mouse right will decrease left axis.

Discharge current can be plotted as a negative value below the X axis or as a positive value above the X axis. See: Menu/System/Use ABS values.

The graph data is cleared each time a new session begins. To print graph see: Menu/File/Print Chart

The Battery Metric device must be connected to PC with Battery Console running in order to collect data for the chart.

5 Using Battery Console

The **Battery Console** software application is used to compile and load a test routine or program into the Battery Metric device. Once the routine has been loaded it is operated using the onscreen controls. A routine/program is a series of steps where each step performs a defined function with defined operating parameters. The routine also has battery details and routing information that allows the routine to advance to other steps when certain conditions are achieved. In this way the routines advance and loop back as required in order to execute the desired battery management task. While the routine is executing the various steps, the software application is polling the device on regular basis to retrieve data and results. This data is displayed on the screen and logged to data files and reports.

How the routine is actually used depends on how the routine is configured to operate. Typically it is set up to be fully automatic. For example, often all you need to do to begin the session is to connect the battery like you would with any battery charger. The device will detect the presence of the battery and begin to execute the routine according to how it was programmed. Often the **START** button is configured to begin the session if the battery is already connected. Sometimes the user will need to know some hints & tips on how to use the routine and what to expect. This information can be found in the Memo section of the routine. Click **Routine Help** button.

Begin use by setting up device with an selected routine or program. There are 3 separate ways this can be done, depending on the features and functionality required.

1. **Quick Setup** – This is a fast way to use default parameters by only entering 3 battery parameters. Type, No of cells, Rated Capacity. This setup method is good when unsure about what parameters to use.
 - Select Quick Setup mode by clicking Quick Setup tab.
 - Enter 3 battery parameters: Battery Type, No of cells, Rated Capacity.
 - Load one of the Standard Routines using the menu/routine selections or equivalent on screen buttons: Charge, Discharge, Cycle.
 - Manage the session using the On screen buttons: Start, Halt, Stop.
 - Standard Routines can also be operated in Stand alone mode without a host PC connection by connecting and disconnecting batteries.
2. **Profile Setup** – This setup method allows for all battery test parameters to be specified as required. This setup method provides for more flexibility when the desired test parameters are known.
 - Select Profile Setup mode by clicking Profile Setup tab.
 - Profiles are saved in a database by name. Select desired profile and edit as required
 - Start profile edits by selecting correct battery type. This will display associated fields.
 - Load one of the Standard Routines using the menu/routine selections or equivalent on screen buttons: Charge, Discharge, Cycle.
 - Manage the session using the On screen buttons: Start, Halt, Stop.
 - Standard Routines can also be operated in Stand alone mode without a host PC connection by connecting and disconnecting batteries.
3. **Program Setup** – Select Program mode by clicking Program tab. Programs are custom routines that are prepared using a XML editor program. See Programming section for more details. When running programs the program will control all aspects of the battery test session. In program mode the Standard Routines do not apply. Standard routines are only available in Quick Setup or Profile Setup mode.
 - Select Program Setup mode by clicking Program tab. This will change the context for the left 3 controls to: Open, Preview, Load.
 - Open the program to make it available for preview and loading.
 - Preview program for inspection.
 - Load program into device for use.
 - Manage the session using the On screen buttons: Start, Halt, Stop.
 - Programs can also be written for use in stand alone mode without a host PC connection.

5.1 Quick Setup

Click Quick Setup tab to enter Quick Setup mode.

Quick Setup		
Required Inputs		
• Battery Type	NiMH	
• No of Cells	4	
• Rated Capacity	1.2	AH
Default Parameters		
• Rated Voltage	4.800	V
• Charge Current	0.240	A
• Discharge Current	0.240	A
• Voltage Cutoff	4.000	V
• No of cycles	2	
• -dV Termination	0.048	
• dT/dt Charge Term	2.000	C/M
QUICK SELECT MODE: Enter the Battery Type, No of Cells and Rated Capacity to use default		

When using Quick Setup mode there are only 3 inputs required to quickly set up the device with default parameters.

1. **Battery Type:** This is the battery chemistry and is usually marked on the battery.
2. **No of Cells:** Total number of cells in the battery. This can usually be determined as follows:
 - NiMH- Rated voltage/1.2
 - NiCd- Rated voltage/1.2
 - SLA- Rated voltage/2
 - LiIon- Rated voltage/3.7
 - Primary- Rated voltage/1.5V
3. **Rated Capacity:** This is usually marked on the battery. Enter value in AH. Estimate if unknown then revise after measuring.

Using these 3 inputs, default parameters will be assigned as follows:

- **Rated Voltage:** Total rated voltage of battery. Total voltage of all cells in series.
 - NiMH- 1.2V/cell
 - NiCd- 1.2V/cell
 - SLA- 2V/cell
 - LiIon- 3.7V/cell
 - LiPO- 3.7V/cell
 - Primary- 1.5V/cell

- **Charge Current:** Rated Capacity / 5
- **Discharge Current:** Rated Capacity / 5
- **Voltage Cutoff:** Lower voltage cutoff value when discharging.
 - NiMH- 1V/cell
 - NiCd- 1V/cell
 - SLA- 1.75V/cell
 - LiIon- 3V/cell
 - LiPO- 3V/cell
 - Primary- 1V/cell
- **No of Cycles:** This is always 2. Once cycle is a charge followed by a discharge
- **-dV Termination:** Terminate charge after peak voltage detection.
 - NiMH- 0.012V/cell
 - NiCd- 0.015V/cell
 - Others: N/A
- **dT/dt Charge Termination:** Terminate charge when rate of temperature reaches this threshold when using optional temperature probe.
 - NiMH- 2°C/Min
 - NiCd- 2°C/Min
 - Others: N/A
- **Min Cur Crg Term:** Terminated charge when current drops below this taper current value during Constant Voltage(CV) charging
 - LiIon- Rated Capacity/10
 - SLA- Rated Capacity/10
 - Others: N/A
- **Float Voltage:** Maintenance voltage that will be maintained when SLA charge session completes.
- **Trickle Current:** Maintenance current that will be maintained when NiCd charge session completes.

After **Quick Setup** inputs have been made the device can be used by loading the device with selected routine and parameters. Loading the routine is done with the **Menu/Routines** selections or using the 3 equivalent on screen controls: **Charge, Discharge, Cycle**. Note that this will not automatically begin a new session. The first 3 controls simply compile and load the routine and parameters into the device. The device is actually controlled by using the other 3 on screen controls: **Start, Halt, Stop**.

Once the routine is loaded into the device the controls-**Start, Halt, Stop** are the only controls that will ever have to be used to control the device and manage the test session. Only if the Quick Setup parameters are changed would the routine have to be loaded again. This is true even when the device is powered off/on.

The following controls can be used once Quick Setup has been set up .



- **Charge:** Load Charge routine into device. This control is not used again unless the parameters have been changed and the routine needs to be reloaded.
- **Discharge:** Load Discharge routine into device. This control is not used again unless the parameters have been changed and the routine needs to be reloaded.
- **Cycle:** Load Cycle routine into device. This control is not used again unless the parameters have been changed and the routine needs to be reloaded.
- **Start:** Route program to Vector step to begin a new session.

- **Halt:** Used to momentarily stop current flow and halt the session. Control will change to **Resume** to continue session
- **Stop:** Used to end the session. Session data is still preserved until a new session begins.

Note that most routines are also configured to begin and end sessions when the battery is connected/disconnected. This way the routines can be used even without the use of the host computer once they have been loaded. The hardware **Vector** push button is equivalent to the on screen **Start** button.

Once the routine is loaded, it does not have to be loaded again, unless any of the test parameter have been changed. This is also true if the device is power off and Battery Console is closed. The next time the device is powered up and Battery Console is restarted the routine can just be used right away without requiring a reload. Just be sure to always connect the correct battery. If a different battery is used or any test parameters need to be changed then reload the routine after making those changes.

Once the routine is loaded, it does not have to be loaded again, unless a different battery type is to be used. This is also true if the device is power off and Battery Console is closed. The next time the device is powered up and Battery Console is restarted the routine can just be used right away without requiring a reload. Just be sure to always connect the same battery type. If a different battery is used then adjust the **Quick Select** battery inputs then reload the routine.

5.2 Profile Setup

Click **Profile Setup** tab to enter Profile Setup mode.




Quick Setup **Profile Setup** **Program**

Profile Name: 12V SLA

Battery Details		
• Battery Type	SLA	
• No of Cells	6	
• Rated Voltage	12	V
• Rated Capacity	6	AH

Notes		
• Battery ID	Generic890	
• Comment	Crimson	

Charge Parameters		
• Charge Current	0.3	A
• Voltage Regulation	14.6	V
• Min Current Term	0.06	A
• Float Voltage	13.5	V
• Maximum Input	0	AH

PROFILE MODE: Select profile from drop down list.   

When using Profile Setup all parameters can be specified as required. Each battery test profile has a unique

name and is stored in the Profile database. Any number of Profiles can be created and they are automatically stored in the database. Use the top drop down selection box to retrieve profiles for use.

The parameters that appear are Battery Type dependent. Start by selecting the battery type so that relevant fields will be displayed. Edit profile as required. Database controls are included to **Create, Copy, Delete** profiles.

- Battery Details: Information about the battery that will be used.
 - **Battery Type** – This is the battery chemistry and is usually marked on the battery. Select from: NiCd, NiMH, SLA,, LiIon, LiPo. For other possible battery types choose one with the same charge methodology.
 - **No of Cells** – No of cells in series. NiMH-Rated V/1.2, NiCd-Rated V/1.2, SLA-Rated V/2, LiIon-Rated V/3.7, Primary-Rated V/1.5V
 - **Rated Voltage** – Total rated voltage of battery. Total voltage of all cells in series.
 - **Rated Capacity** – This is usually marked on the battery. Enter value in AH. Estimate if unknown then revise after measuring.
- Notes: Notes that will appear on the screen and in reports
 - **Battery ID** – Battery Identification such as model or serial number.
 - **Comment** – Any comment that is wanted on the printed report.
- Charge Parameters: Items used to manage charging and charge termination
 - **Charge Current(A)** – For constant current or initial current limit.
 - **Voltage Regulation(V)** – Constant voltage limit for SLA & LiIon only.
 - **Min Current Term(A)** – Taper current value to terminate charge for SLA & LiIon only.
 - **Charge Time Limit(Min)** – Terminate charge after this amount of time.
 - **Maximum Input(AH)** – Charge will terminate when this level of input is achieved.
 - **-dV Termination(V)** – Terminate charge after peak voltage detection for NiMH, NiCd only.
 - **dT/dt Termination(C/Min)** – Terminate charge when this rate of temperature rise is detected with the optional temperature probe. For NiMH, NiCd only.
 - **Trickle Current(A)** – For NiCd only, apply a small maintenance current when charge completes.
 - **Float Voltage(V)** – For SLA only, apply a maintenance voltage when charge completes.
- Discharge Parameters: Items used to manage discharging and discharge termination
 - **Discharge Current(A)** – Constant current discharge value.
 - **Low Voltage Cutoff(V)** – Discharge terminates when voltage drops to this value.
 - **Maximum Output(AH)** – Discharge will terminate when this amount of output is delivered.
 - **Discharge Time Limit(Min)** – Terminate discharge after this amount of time.
 - **Pass Threshold(%)** – Value used to calculate and display pass/fail messages.
 - **Pulse1 Current(A)** – Specifies first part of pulse. Use 0 to disable pulsing.
 - **Pulse1 Duration(Sec)** – Interval for first part of pulse.
 - **Pulse2 Current(A)** – Specifies second part of pulse. Use 0 to disable pulsing.
 - **Pulse2 Duration(Sec)** – Interval for second part of pulse.
- Options: Optional parameters that can be used to manage test sessions
 - **No of Cycles** – Number of Charge-Discharge cycles desired.
 - **High Voltage Cutoff(V)** – Can be used to terminate a step when exceeded.
 - **Temp Cutoff(C)** – Terminate step at this temperature when using the optional temperature probe.
 - **Pause after Charge(Min)** – Use a pause interval after charge to allow battery to cool or rest.
 - **Pause after Discharge(Min)** – Use a pause interval after discharge to allow battery to cool or rest.
 - **Group Size** – Specify group size for current sharing.

After **Profile Setup** inputs have been made the device can be used by loading the device with selected routine and parameters. Loading the routine is done with the **Menu/Routines** selections or using the 3 equivalent on screen controls: **Charge, Discharge, Cycle**. Note that this will not automatically begin a new session. The first 3 controls simply compile and load the routine and parameters into the device. The device is actually controlled by using the other 3 on screen controls: **Start, Halt, Stop**.

Once the routine is loaded into the device the controls-**Start, Halt, Stop** are the only controls that will ever have to be used to control the device and manage the test session. Only if the Profile Setup parameters are changed would the routine have to be loaded again. This is true even when the device is powered off/on.

The following controls can be used once Quick Setup has been set up .



- **Charge:** Load Charge routine into device. This control is not used again unless the parameters have been changed and the routine needs to be reloaded.
- **Discharge:** Load Discharge routine into device. This control is not used again unless the parameters have been changed and the routine needs to be reloaded.
- **Cycle:** Load Cycle routine into device. This control is not used again unless the parameters have been changed and the routine needs to be reloaded.
- **Start:** Route program to Vector step to begin a new session.
- **Halt:** Used to momentarily stop current flow and halt the session. Control will change to **Resume** to continue session
- **Stop:** Used to end the session. Session data is still preserved until a new session begins.

Note that most routines are also configured to begin and end sessions when the battery is connected/disconnected. This way the routines can be used even without the use of the host computer once they have been loaded. The hardware **Vector** push button is equivalent to the on screen **Start** button.

Once the routine is loaded, it does not have to be loaded again, unless any of the test parameter have been changed. This is also true if the device is power off and Battery Console is closed. The next time the device is powered up and Battery Console is restarted the routine can just be used right away without requiring a reload. Just be sure to always connect the correct battery. If a different battery is used or any test parameters need to be changed then reload the routine after making those changes.

5.3 Program Setup

Click **Program** tab to enter Program mode.

Quick Setup | **Profile Setup** | **Program**

Filename: MyCycleLilon2A.xml

Title: Cycle Lilon

Battery Type: lilon

Description: Cycle Generic Lilon Battery

Rating: 3.7V / 2.4AH

Cells: 1-Ser / 1-Par

Reset Step: 3

Default Vector: 2

Group Size: 0

MEMO: This function is used to cycle a generic Lilon battery for the specified number of cycles. Specify charge and discharge parameters accordingly. Remove and reconnect battery to begin

Programs are custom routines that are prepared using a XML editor program. See Programming section for more details.

Open Preview Load Start Halt Stop

When in Program mode the function of the 3 leftmost control buttons on the main screen change functions. These 3 control buttons now are used to *Open*, *Preview* and *Load* program into the device for use. The *Standard Routines* are not use in *Program Mode*. When a program is opened, the Program Setup window area displays some operational information about the program. The complete program listing can be viewed using Preview.

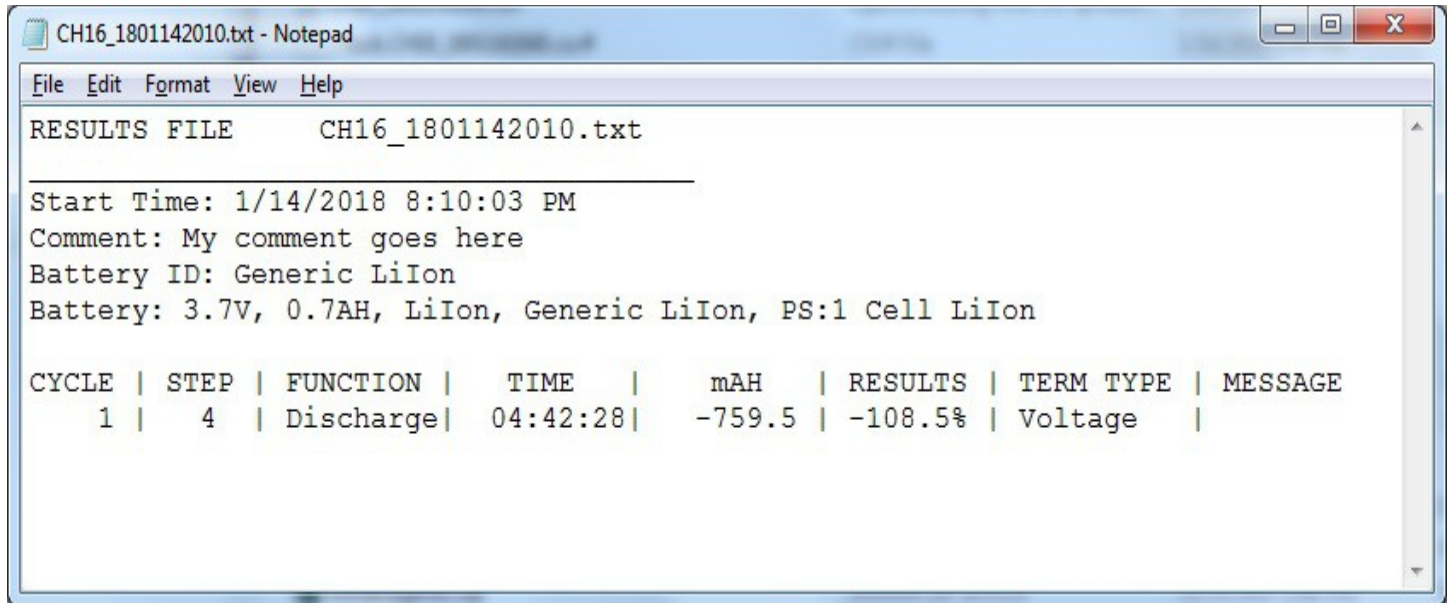
Note that Programs can be configured to begin and end sessions when the battery is connected/disconnected. This way the program can be used even without the use of the host computer once they have been loaded. The hardware **Vector** push button is equivalent to the on screen **Start** button. See **Programming** section for details on creating programs.

Once the program is loaded, it does not have to be loaded again, unless a different program is required. This is also true if the device is powered off and Battery Console is closed. The next time the device is powered up and Battery Console is restarted the program can just be used right away without requiring a reload. Just be sure to always connect the correct battery for the programmed session. If a different battery is used or any test parameters need to be changed then reload of a new program will be required.

5.4 Data Collection

In addition to the Real Time Data and Results Data that appears on the main screen, users also have the ability to create two data files that collect data during a session. These are Results logging and Data logging.

Results Logging



The **Results Logging** file creates one line entry with the results for each step that completes during a session. A new **Results Logging** file is created every time a new session begins. The new session is triggered by executing the Reset step in the routine. When this new session begins a new file will be created using the same file location path as specified in the **Data Logging** menu item. This **Results file** will have the format:

CHX_YYMMddHHmn.txt

X=Channel No YY=Year MM=Month

dd=Day HH=Hour mn=Minute

Enable or disable the creation of the **Results Logging** file by clicking the menu item: **System/Results Logging**. When each step completes a one line update is made to the Results Logging file. This update is similar to what you would see on one line in the Saved Data section. However by using this file you can have a permanent record of the results and also collect more results than can fit in the Saved Data section. For example when doing battery life cycle testing for hundreds of cycles the results file will collect and store data from each cycle. **Note:** Unlike the **Results Data** on the main screen, the Battery Console application must be running in order to make updates to the **Results Logging** file.

The Results Logging file is a plain text file that can be opened and viewed with any text editor program. If you open this text file with a text editor, like Wordpad, during a session, it will lock the file and prevent new updates. In this case copy the file and only open the new copy. You may be able to view an open file using a text editor in read only mode. Once a new session begins the last text file is closed and available for viewing.

Stop steps are not recorded in the **Results Logging** file.

Data Logging

Log#	Step#	Count1	Function	StepTime(Min)	TotalTime(Min)	Voltage(V)	Current(mA)	Power(W)	Capacity(AH)	Energy(WH)	%Cap(AH)	IntRes(mΩ)	Temp(C)
1	1	4	1 Discharge	0.033	0.033	3.995	-162	-0.647	-0.1	0	0---	0	0
2	2	4	1 Discharge	1.047	1.047	3.974	-162	-0.644	-2.8	0	-0.4---	0	0
3	3	4	1 Discharge	2.061	2.061	3.961	-162	-0.642	-5.5	0	-0.8---	0	0
4	4	4	1 Discharge	3.075	3.075	3.952	-162	-0.64	-8.3	0	-1.2---	0	0
5	5	4	1 Discharge	4.089	4.089	3.944	-162	-0.639	-11	0	-1.6---	0	0
6	6	4	1 Discharge	5.103	5.103	3.937	-162	-0.638	-13.7	-0.1	-2---	0	0
7	7	4	1 Discharge	6.117	6.117	3.935	-162	-0.637	-16.5	-0.1	-2.4---	0	0
8	8	4	1 Discharge	7.131	7.131	3.935	-162	-0.637	-19.2	-0.1	-2.7---	0	0
9	9	4	1 Discharge	8.145	8.145	3.935	-162	-0.637	-21.9	-0.1	-3.1---	0	0
10	10	4	1 Discharge	9.159	9.159	3.935	-162	-0.637	-24.6	-0.1	-3.5---	0	0
11	11	4	1 Discharge	10.173	10.173	3.935	-162	-0.637	-27.4	-0.1	-3.9---	0	0
12	12	4	1 Discharge	11.187	11.187	3.935	-162	-0.637	-30.1	-0.1	-4.3---	0	0
13	13	4	1 Discharge	12.201	12.201	3.935	-162	-0.637	-32.8	-0.1	-4.7---	0	0
14	14	4	1 Discharge	13.215	13.215	3.935	-162	-0.637	-35.5	-0.1	-5.1---	0	0
15	15	4	1 Discharge	14.229	14.229	3.935	-162	-0.637	-38.3	-0.2	-5.5---	0	0
16	16	4	1 Discharge	15.243	15.243	3.935	-162	-0.637	-41	-0.2	-5.9---	0	0
17	17	4	1 Discharge	16.257	16.257	3.831	-161	-0.617	-43.8	-0.2	-6.3---	0	0
18	18	4	1 Discharge	17.271	17.271	3.839	-161	-0.618	-46.5	-0.2	-6.6---	0	0
19	19	4	1 Discharge	18.285	18.285	3.835	-161	-0.617	-49.2	-0.2	-7---	0	0
20	20	4	1 Discharge	19.299	19.299	3.834	-161	-0.617	-51.9	-0.2	-7.4---	0	0
21	21	4	1 Discharge	20.313	20.313	3.832	-161	-0.617	-54.6	-0.2	-7.8---	0	0
22	22	4	1 Discharge	21.327	21.327	3.829	-161	-0.616	-57.4	-0.2	-8.2---	0	0
23	23	4	1 Discharge	22.341	22.341	3.826	-161	-0.616	-60.1	-0.2	-8.6---	0	0

The Data Logging file records the following real time data at specified intervals:

Header	Log#	Step#
Count1	Function	StepTime(Min)
TotalTime(Min)	Voltage(V)	Current(mA)
Power(W)	Capacity(AH)	Energy(WH)
%Cap(AH)	IntRes(mΩ)	Temp(C)

Set the Data Logging interval and file path location by clicking the menu item: **System/Results Logging**

Any interval greater than zero will enable the data logging. Set the interval to zero if data logging is not required.

This **Data Logging** file will have the format: **CHX_YYMMddHHmn.csv**

X=Channel No YY=Year MM=Month

dd=Day HH=Hour mn=Minute

The Battery Console application must be running in order to make updates to the **Data Logging** file. The Data Logging file uses a Comma Separated Variable (CSV) format. This allows for easy importing to popular spreadsheet programs such as Excel or Open Office.

Download example Data Log file: [CH16_1801142010.csv](#)

For reference the data logging interval is shown in the bottom status bar on the main window.

6 Standard Routines

The *Battery Console* application has a selection of pre-programmed standard routines that can be used as required. No programming is required to use these routines. Simply select how you want to define the parameters and then load any of these standard routines into the *Battery Metric* device for use. There are two ways to define the test parameters prior to loading these standard routines:

1. **Quick Setup** – Simply select the *Quick Setup* tab in the setup panel to enter *Quick Setup* mode. Then enter the 3 required parameters: *Battery Type*, *Number of cells* in series and the *Rated Capacity*. Default parameters will be used based on these 3 inputs. If the rated capacity is unknown then just use the best estimate and use the device to measure the actual capacity. Revise this value later with the new information.
2. **Profile Setup** – Select the *Profile Setup* tab in the setup panel to enter *Profile Setup* mode. In profile mode a database holds all of the battery setup profiles. Each profile has a name and associated parameters. Create new profiles or edit the existing profiles as required. The software comes with a few example profiles. These example Profiles should be reviewed and edited as required for the correct battery being managed.

Standard routines can only be used in *Quick Setup* or *Profile Setup* mode. The 3 leftmost onscreen controls: *Charge*, *Discharge*, *Cycle* are used to load these respective routines into the device. These 3 standard routines along with additional standard routines can also be found using: *Menu/Routines*.

Note that in *Program Setup* mode these standard routines are no longer available. The 3 onscreen controls: *Charge*, *Discharge*, *Cycle* will be replaced with program management controls when in *Program Setup* mode.

6.1 Charge

Standard Charge Routines

The Standard Charge Routines will charge batteries using a charge algorithm appropriate for the specified battery type. Specify *Battery Type* in either *Quick Setup* or *Profile Setup* mode. In *Quick Setup* mode, default battery charge parameters will be used. Profile Setup allows for the charge parameters to be specified as required.

After setting up the charge parameters click the on screen **Charge** button. This will load the appropriate charge routine into the device. Once the routine has been loaded, it is controlled with the on screen, **Start**, **Halt**, **Stop** buttons. The charge routines are also configured to begin the session simply by connecting the battery. The hardware Vector push button can also be used to restart the charge session. Note that after the charge routine is loaded into the device, the Charge button is not used again unless the test parameters need to be updated. Also, once the charge routine is loaded, the device can also be used off line without the PC connection for stand alone applications.

Charge SLA

SLA batteries charge with a CC,CV algorithm. Specify maximum initial current as *Charge Current* and the maximum voltage as *Voltage Regulation*. The battery will charge at Constant Current until the battery voltages rises to the maximum voltage. Once the battery voltage reaches the maximum specified, the device will switch to Constant Voltage regulation. In CV regulation the current will taper down as the battery charging process

continues. The battery will be fully charged when the current tapers down below the *Min Current Term* value specified. Once the charge terminates a maintenance float voltage will be applied if the optional *Float Voltage* value is specified.

Memo: This function is for charging a generic SLA battery. Remove and reconnect battery to begin a new session. Flashing green LED indicates charging. Once fully charged a solid green LED indicates that a float voltage is being applied for maintenance until the battery is disconnected.

Charge LiIon

LiIon are charged using CC/CV method

Memo: This function is for charging a generic LiIon battery. Specify charge parameters accordingly. Remove and reconnect battery to begin a new session. Flashing green LED indicates charging. A solid green LED indicates charging is complete.

Charge LiPO

Memo: This function is for charging a generic LiPO battery. Specify charge parameters accordingly. Remove and reconnect battery to begin a new session. Flashing green LED indicates charging. A solid green LED indicates charging is complete.

Charge NiMH

Memo: This function is for charging a NiMH battery. Remove and reconnect a battery to begin a new session. Flashing green LED indicates charging. A solid green LED indicates that charging is complete. An under voltage and over voltage situation will be detected as a possible incorrect battery. Check battery or Restart/Vector to force the charge to begin. Charge will terminate for safety at 130% of capacity. Users should also include additional termination methods in the battery profile. Recommended termination methods are -dV of 0.012V/cell, Maximum input of 120%, Temperature cut-off of 40 deg C and dT/dt of 2 deg C/Min. These should be adjusted as battery, charge rate, temperature and situation dictates. Always charge batteries as per the battery manufacturers' recommendations.

Charge NiCd

NiCd is charged using Constant Current(CC).....

Memo: This function is for charging a NiCd battery. Remove and reconnect a battery to begin a new session. Flashing green LED indicates charging. Specify a Trickle Current value (>0.005A) in the Profile to automatically apply a trickle current once fully charged. A solid green LED indicates that the specified trickle current is being applied for maintenance until the battery is disconnected. An under voltage and over voltage situation will be detected as a possible incorrect battery. Check battery or Restart/Vector to force the charge to begin. Charge will terminate for safety at 140% of capacity. Users should also include additional termination methods in the battery profile. Recommended termination methods are -dV of 0.015V/cell, Maximum input of 120%, Temperature cut-off of 40 deg C and dT/dt of 2 deg C/Min. These should be adjusted as battery, charge rate, temperature and situation dictates. Always charge batteries as per the battery manufacturers'

recommendations.

6.2 Discharge

Standard Discharge Routine

The *Discharge* routine will discharge a battery down to the *Voltage Cutoff* value specified in *Quick Setup* or *Profile Setup*. The output delivered will be measured in mAH, or AH and WH. A rating of *Percent Capacity* will also be calculated for the battery. This rating will be based on the actual AH output delivered expressed as a percentage of the rated capacity. Be sure to specify the Rated Capacity in order for this rating to be representative. By specifying a pass threshold a *Pass* or *Fail* text message will also be calculated at the end of the discharge session. This text message will appear in the *Results Data* when the discharge session terminates.

Memo: Universal Discharge Function. Connect battery to begin session. Battery will discharge down to lower cut-off voltage specified. Flashing yellow LED indicates the test is proceeding. Solid green LED indicates output exceeded pass threshold. Solid yellow LED indicates battery failed to deliver the pass value.

Pulse Discharge Option

A two part discharge pulse can be specified rather than a constant current discharge.

Quick Setup **Profile Setup** **Program**

Profile Name: 1 Cell Lilon

Discharge Parameters		
• Discharge Current	0	A
• Low Voltage Cutoff	3	V
• Maximum Output	0	AH
• Discharge Time Limit	0	Min
• Pass Threshold	80	%
• Pulse 1 Current	0.05	A
• Pulse 1 Duration	10	Sec
• Pulse 2 Current	0.75	A
• Pulse 2 Duration	0.25	Sec

Options

• No of Cycles	4
----------------	---

PROFILE MODE: Select profile from drop down list.

Icons: +, →, X

To implement this feature set the *Discharge Current* = 0 and specify the pulse requirements. There are four fields used to define a two part pulse. Specify the *Current* and *Duration* for each part. The two parts of the pulse will repeat continuously for the duration of the discharge step. In the example above, the battery will discharge

at 50mA with a 750mA pulse for 250mS every 10 seconds.

For a more complex pulse, a custom routine can be programmed. Custom routines can define a pulse with up to 6 parts. See *Program Pulsing* for more information.

When the pulsing feature is not being used, set all four pulse fields = 0 and specify a constant current discharge using *Discharge Current*.

6.3 Cycle

Standard Cycle Routines

The Standard Cycle Routines will cycle the batteries for the number of cycles specified in the *No of Cycles* parameter. Each cycle consists of a discharge followed by a charge. For example if the *No of Cycles* = 2 then the cycle session will consist of: Cycle 1 Discharge, Cycle 1 Charge, Cycle 2 Discharge, Cycle2 Charge.

Specify *Battery Type* in either *Quick Setup* or *Profile Setup* mode. In *Quick Setup* mode, default cycle parameters will be used. Profile Setup allows for the cycle parameters to be specified as required. The parameter *Discharge Current* will set the constant current discharge level and *Voltage Cutoff* will specify when to terminate the discharge. The charge cycles will charge batteries using a charge algorithm appropriate for the specified battery type.

During a cycle session the actual battery capacity can be measured on the Discharge portion of the cycle. This is when the actual output of the battery is determined. At the end of each part of the cycle, the results are displayed in the *Results Data* area. If the battery was fully charged at the beginning of the cycle session, then the first discharge will reveal the total capacity of the battery. If the state of charge was less than 100% then the first discharge will not represent the full capacity. On the second discharge after the battery has been charged to 100% the full capacity will be measured. The cycle session ends with a charge so that the battery will be ready for service when complete.

After setting up the cycle parameters click the on screen **Cycle** button. This will load the appropriate cycle routine into the device. Once the routine has been loaded, it is controlled with the on screen, **Start**, **Halt**, **Stop** buttons. The cycle routines are also configured to begin the session simply by connecting the battery. The hardware Vector push button can also be used to restart the cycle session. Note that after the cycle routine is loaded into the device, the *Cycle* button is not used again unless the test parameters need to be updated. Also, once the cycle routine is loaded, the device can also be used off line without the PC connection for stand alone applications.

Cycle NiCd

Memo: This function use to cycle a NiCd battery for the specified number of cycles. Remove and reconnect a battery to begin a new session. Flashing yellow LED indicates discharging. Specify Low Voltage cutoff to terminate discharge. Flashing green LED indicates charging. A solid green LED indicates that the cycles are complete and a trickle current(if specified>0.005A) will be applied until the battery is disconnected. An under voltage and over voltage situation will be detected as a possible incorrect battery. Check battery or Restart/Vector to force the charge to begin. Charge will terminate for safety at 140% of capacity. Users should also include additional charge termination methods in the battery profile. Recommended terminations are -dV of

0.015V/cell, Maximum input of 120%, Temperature cut-off of 40 deg C and dT/dt of 2 deg C/Min. These should be adjusted as battery, charge rate, temperature and situation dictates. Always charge batteries as per the battery manufacturers recommendations.

Cycle Lilon

Memo: This function is used to cycle a generic Lilon battery for the specified number of cycles. Specify charge and discharge parameters accordingly. Remove and reconnect battery to begin a new session. Each cycle consists of a discharge followed by a charge. Flashing yellow LED indicates discharging. Flashing green LED indicates charging. A solid green LED indicates cycles are complete.

Cycle NiMH

Memo: This function is for charging a NiMH battery. Remove and reconnect a battery to begin a new session. Flashing green LED indicates charging. A solid green LED indicates that charging is complete. An under voltage and over voltage situation will be detected as a possible incorrect battery. Check battery or Restart/Vector to force the charge to begin. Charge will terminate for safety at 130% of capacity. Users should also include additional termination methods in the battery profile. Recommended termination methods are -dV of 0.012V/cell, Maximum input of 120%, Temperature cut-off of 40 deg C and dT/dt of 2 deg C/Min. These should be adjusted as battery, charge rate, temperature and situation dictates. Always charge batteries as per the battery manufacturers' recommendations.

6.4 Recover

Standard Recover Routine

Use this routine to recover deeply discharged batteries.

Memo: Some routines will not proceed when the battery voltage is below limits for the battery type and number of cells. This routine can be used to apply the specified "Charge Current" to the battery regardless of initial terminal voltage. This could be used to recover a dead battery for evaluation purposes. This routine is not battery type specific and will automatically terminate after 5 minutes for safety. Specify a "High Voltage Cutoff"(Pass) or "Max Charge Time"(fail) to terminate earlier. At this point, the other regular routines could then be used if the battery voltage rises within bounds.

6.5 Load Test

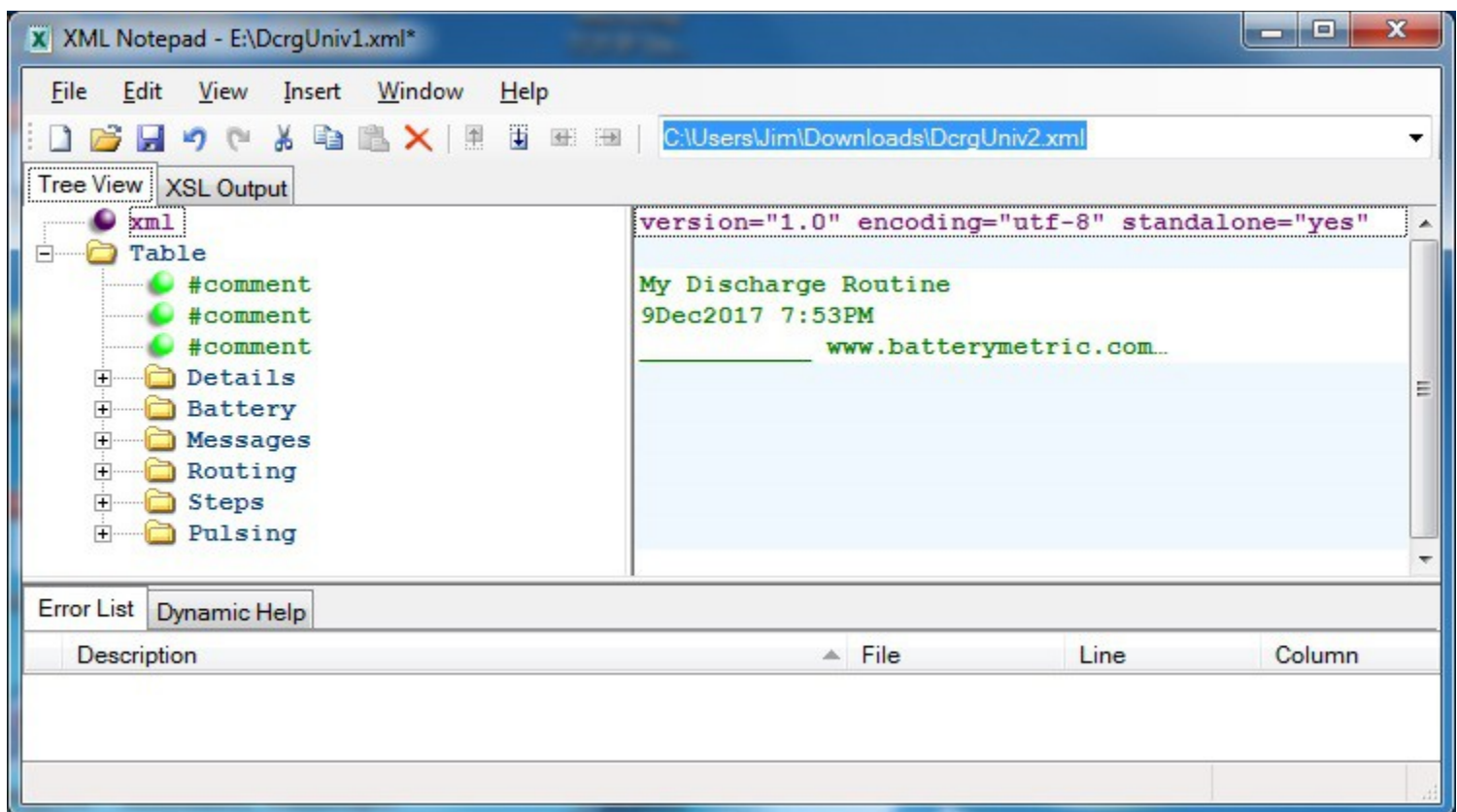
Standard Load Test Routine

The load test applies a discharge current load to see if the battery can maintain an output terminal voltage while loaded. This can be a quick test with a *Pass* or *Fail* indication about battery health.

Memo: Use this routine to apply a load to test if a battery can maintain a voltage for a specified amount of time. Specify test time as “Discharge Time Limit” parameter. Alternatively specify Maximum AH Output. Specify pass voltage as “Low Voltage Cutoff” parameter. Specify current as “Discharge Current” parameter. Connect battery to begin session. Battery will discharge until the time specified. If the voltage drops below the “Low Voltage Cutoff” the test will immediately stop and register a Fail. Flashing yellow LED indicates the test is proceeding. Solid green LED(Pass) indicates the voltage was maintained above the threshold specified. Solid yellow LED(Fail) indicates battery failed to maintain voltage above threshold. Repeat for any number of batteries. Once the routine is loaded it can operate without connection to PC.

7 Programming

Programs are structured files that contain all of the details, step and routing instructions that define how a routine will operate. The program operates by moving from one step to another as defined by the Routing Statements.



Programs are written and developed as XML files. In order to program custom routines you will first need to create a new blank template. See: **Menu/Program/New**

Once the template is created it can be edited using any XML editing program. The recommended program for this is XML Notepad. This is a free editing tool from Microsoft. Download here: [XML Notepad](#)

There are 6 sections in the XML file where program information and instructions are stored.

Details: Overall Program settings

Battery: Information about the battery

Messages: Custom Messages to user

Routing: Routing statements that control operation

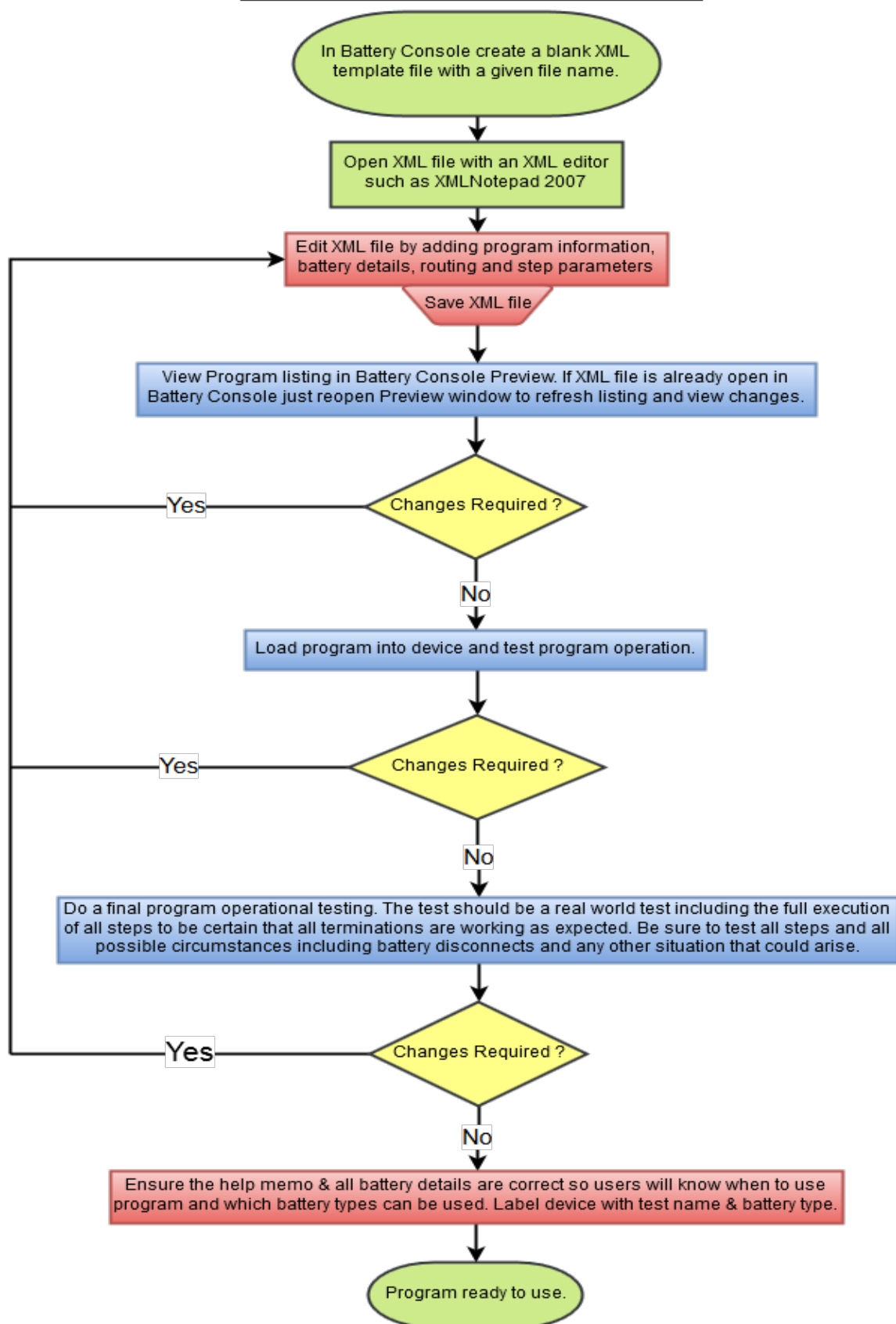
Steps: Parameters for each step

Pulsing: Pulsing definitions

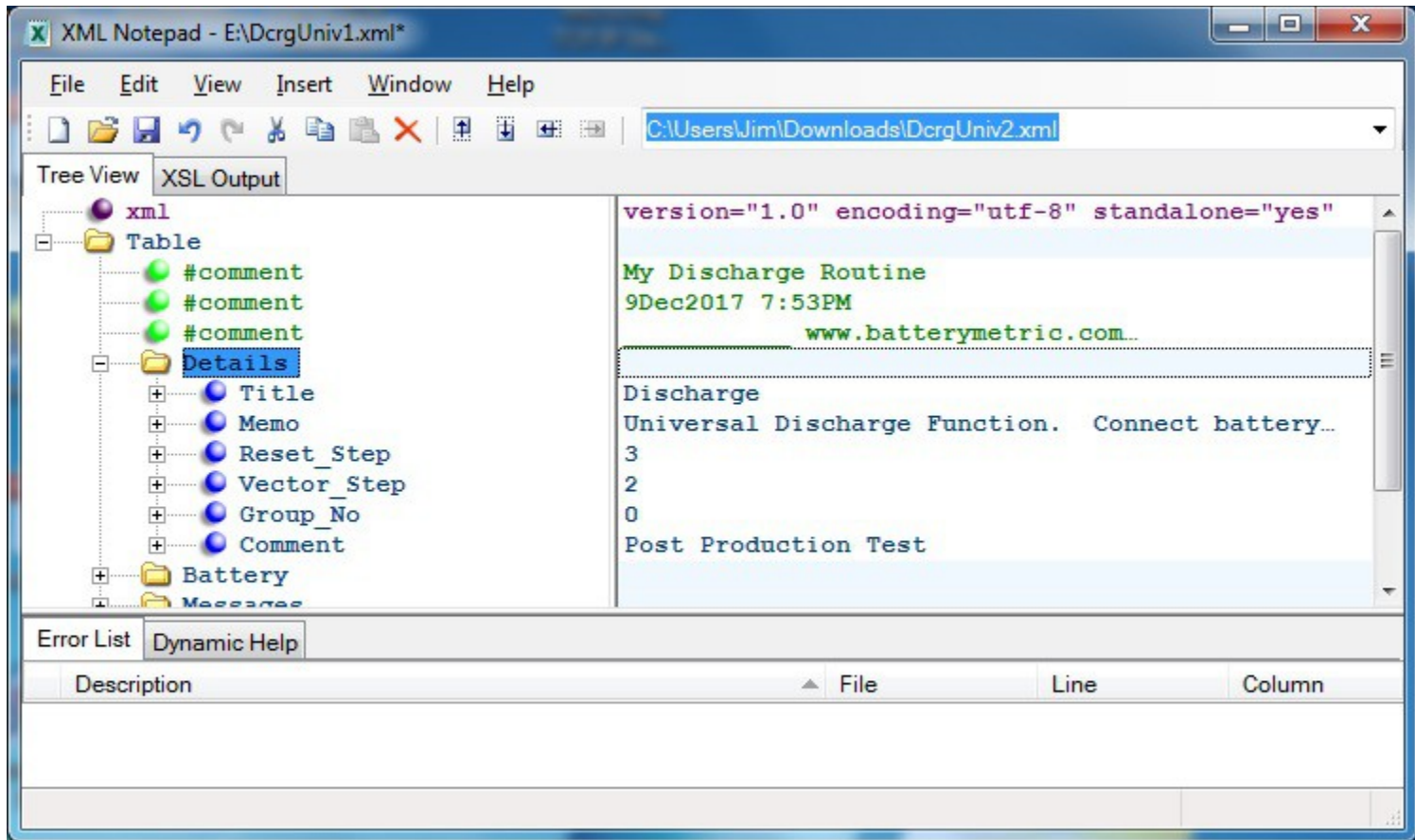
Editing an existing program is straight forward. Just open the xml file with the xml editor, edit the file as required and then save. Once the file is saved it can be opened in Battery Console for **Preview** or Loading into the device. For convenience, during program development and debugging, the Battery Console application includes a **Preview** window that will provide a comprehensive listing of the program. This **Preview** window displays the program details in an easy to read format for development, review and debugging. If the Program is already open in Battery Console, when a change is made to the XML file using the editor, then just **save** the file in the XML editor then reopen the **Preview** window to refresh the program view.

Developing a new program is an iterative process:

Program Development Workflow



7.1 Program Details



Title Name for this routine. It will appear in the title bar of the main screen when used.

Memo This is helpful information for the user's reference. Include things like the purpose of the routine, how to use it, how it works, important application information or battery details. This memo data gets loaded into the device so the user can refer to this memo by click the “ ? ” icon. This memo also appears on printed reports. The maximum length of the memo depends on how many steps are used. See **Memory Organization** section for more details. Generally 1000+ characters can be used.

Reset_Step This is the step number that will erase saved data from the previous session and begin a new battery session.

Vector_Step This is the default vector step. The routine will route to the default vector step whenever the Restart button is clicked or the hardware push button is depressed on the device. The only exception is if a Vector step is defined (>0) in the currently operating step. In which case all Vector requests will route to the step defined Vector.

Group_No This is the number of channels in the Current Sharing group. See the **Current Sharing** section for more details. Keep this value = 0 unless using the current sharing option is being used.

Comment This appears on printed reports. It could be used to add a comment, author name, phone number or company name. Maximum of 32 characters.

The programming information in the XML file directs the program operation. The XML file also contains

information to inform the user about what is happened at any given time and overall information about what the program is for and what type of battery should be used. This ensures that there is no misunderstanding by the user as to how and when to use the program and which type of battery should be connected to the device. Be sure to accurately reflect all these details in the program XML file.

The Reset and Vector Steps

Reset Step: Whenever the program begins this step number it will clear all data from the previous session and start a new session. By carefully considering which step should perform a Reset, you can be in complete control over when the last session ends and the new one begins. The Reset Step must be a number between 1 – 64

Vector Step: This is the step the program will route to when a user presses the on screen “Restart” or hardware “Vector” push switch. In all the Standard Routines the default Vector Step = 2. The default Vector step assignment can get over-ridden in any particular step by specifying an alternate value (>0). This creates a lot of possibilities for creating special test routines. This would allow for programs that perform multiple task and the operator could switch between tasks using the Vector or Restart buttons. See **Programming Strategies** section in this manual for more ideas on how to use this. The Vector Step can be a number between 0 – 64 where 0 means the Vector function is disabled.

Considerations for configuring the initial steps a program

The Standard Routines use a Reset Step = 3 and a Vector Step = 2 because that is the best way to manage the power on event and preserve the data from that last session. There are several considerations that make this a good choice when developing new programs:

- The Battery Metric devices are designed to operate in stand alone mode. Once the program is loaded into the device it can be used simply by connecting a battery. When a session is complete, just remove the battery and connect another.
- When powering up the device we want to preserve the Saved Data results from the last session.
- In the event the battery session was performed without using Battery Console then we want the option to be able to Power Up device then Open Battery Console and see the results from the last session.
- It's not a good idea to have the new session begin as soon as the device is powered up because there could be a battery connected inadvertently. It's wise to be able to power the device up even if a battery is connected without a session starting. This way there is still opportunity after power up to view last session, or have a chance to remove the battery before deciding which battery to connect next.
- The device needs to manage unexpected power downs. This could include a power failure or accidental disconnect. Whenever power is lost it may be undesirable to have the session restart immediately upon restoration. This is why the first step waits for the battery to be disconnected before proceeding. There needs to be a way for the user to decide if they want to view data up to the last power failure, or restart the session on the same battery or take other action. Also if there is no operator present at the time of power loss/restoration the device should wait for user intervention before proceeding.

Sample program start listing as viewed in the Preview window

FIRST POWER ON STEP

```
S1:Wait for battery to be removed
  Function:stop, Vreg:0V, Ireg:0A, Instruction:Remove Battery
  R1:(term)If voltage < 0.5 GoTo Next    (wait until battery removed)
```

DEFAULT VECTOR STEP

```
S2:Wait for battery to be connected
  Function:stop, Vreg:0V, Ireg:0A, Instruction:Connect correct battery to begin
  R2:(term)If voltage >= 0.5 GoTo Next    (Proceed when battery connected)
```

SESSION RESET STEP

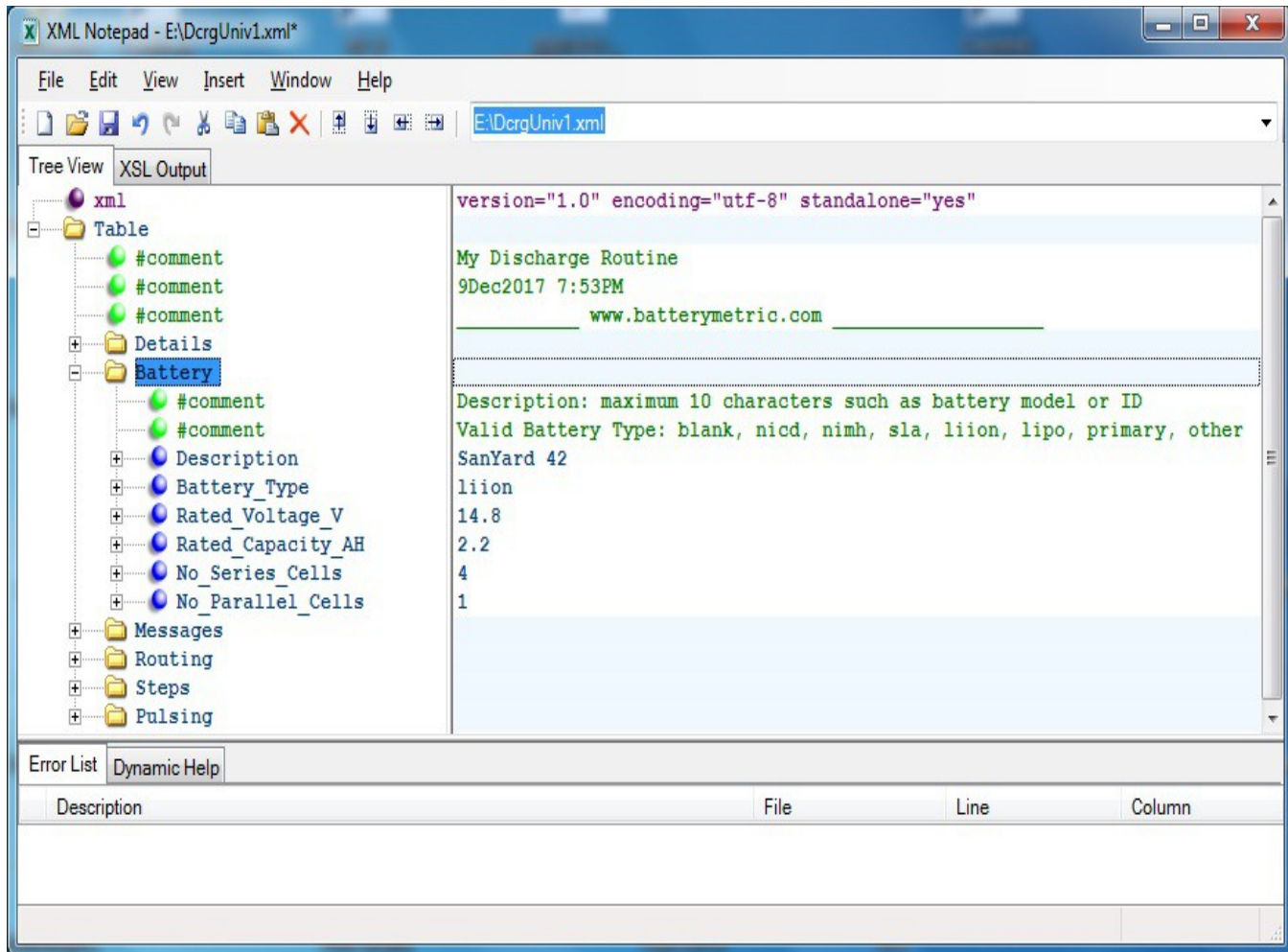
```
S3:Start new session
  Function:pause, Vreg:0V, Ireg:0A, Instruction:Wait...
  R10:(term)If voltage > 4.75 GoTo 9      (Over voltage/wrong battery)
  R11:(term)If voltage >= 0.5 GoTo Next, Inc Count1 (Continue to next step)
```

Note: The onscreen “**Restart**” button forces program to the Vector step. This way if a battery is already connected then the program just proceeds to step 3 where the reset occurs and the new session can immediately begin. If no battery is connected using “Restart” then program will just wait and automatically restart when the battery is connected. However if the presently operating step has a Vector specified (>0) then this step Vector will take precedence.

The onscreen “**Stop**” button always forces the program to step 1. The program will remain on step 1 as long as the battery is connected, giving the user a chance to see the Save Data that has been collected. Only when the battery is physically disconnected & reconnected or user presses “Restart” will a new session begin.

Safety First: The on screen *Stop* button forces the routine to step 1. This step should be configured to stop all charge and discharge current for situations when the user may need to perform an emergency stop.

7.2 Program Battery



Description

Enter any type of battery identification, like model number or battery type. Maximum of 10 characters. This will be displayed in section 3 on the Main Screen so users will know exactly what type of battery is to be used. It also appears on printed reports.

Battery_Type

Use one of the valid battery chemistry qualifiers. Use the exact syntax. Since this is used for validation during program compiling you must use one of the following:
nicd, nimh, sla, liion, lipo, primary, other

This is the total rated voltage of all cells in series. Typical values:

Rated_Voltage_V

- nicd & nimh: No of cells x 1.2V/cell
- sla: No of cells x 2V/cell
- liion & lipo: No of cells x 3.7V/cell

Rated_Capacity_AH

Capacity rating of the battery specified in Amp Hours(AH). Used to perform % of Capacity calculations. Estimate if unknown and then revise once it's been measured.

No_Series_Cells

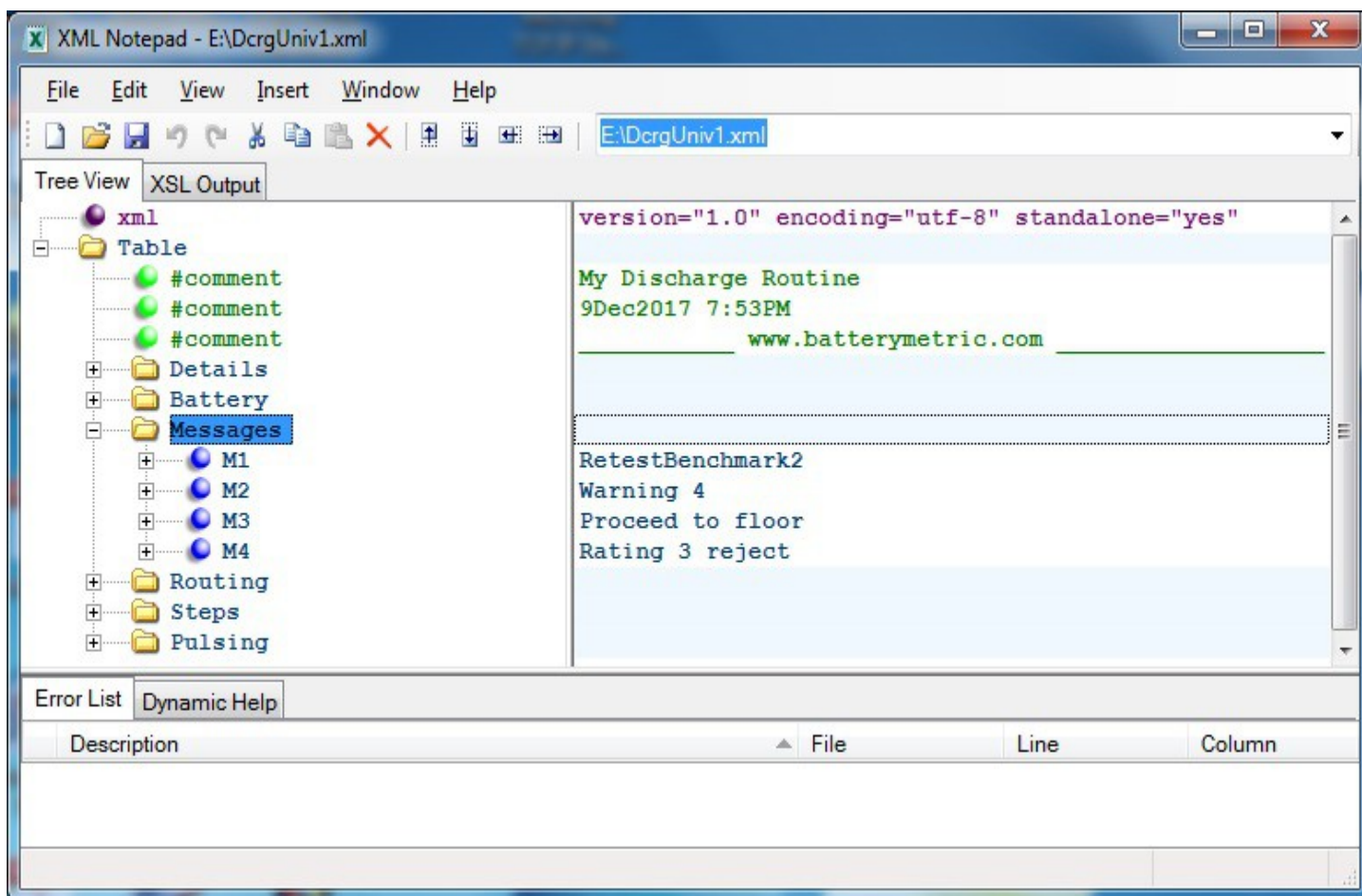
No of cells in series. For single cell value = 1. Calculate for packs using V/cell listed above.

No_Parallel_Cells

No of cells in parallel for information reports. If unknown use 1

Battery information entered here appears in various places when the program is loaded. This is important so users will understand when and how to use the program. Before connecting any batteries, users should check this on screen information to ensure they are connecting the correct battery.

7.3 Program Messages



Messages are used in Battery Console to keep the user informed and provide instructions. They appear in Section 3 of the main screen where they display an information message to the user as defined in the Program Step that is currently operating.

Messages also appear in the Saved Data results and convey information about the step that has completed. Messages will also appear in other areas such as printed reports and data files. By using the routing statements, the Saved Data messages can be based on calculations on the measured parameters determined during the step. Example Routing statement:

- mess: if mAH \geq value then GOTO 29 (Pass)
- mess: if mAH $<$ value then GOTO 30 (Fail)

where GOTO represents the defined message number.

This is just one example. All of the test parameters can be examined and appropriate messages calculated.

Select from any of the predefined messages as shown in the following Message Number Listing.

Custom Messages

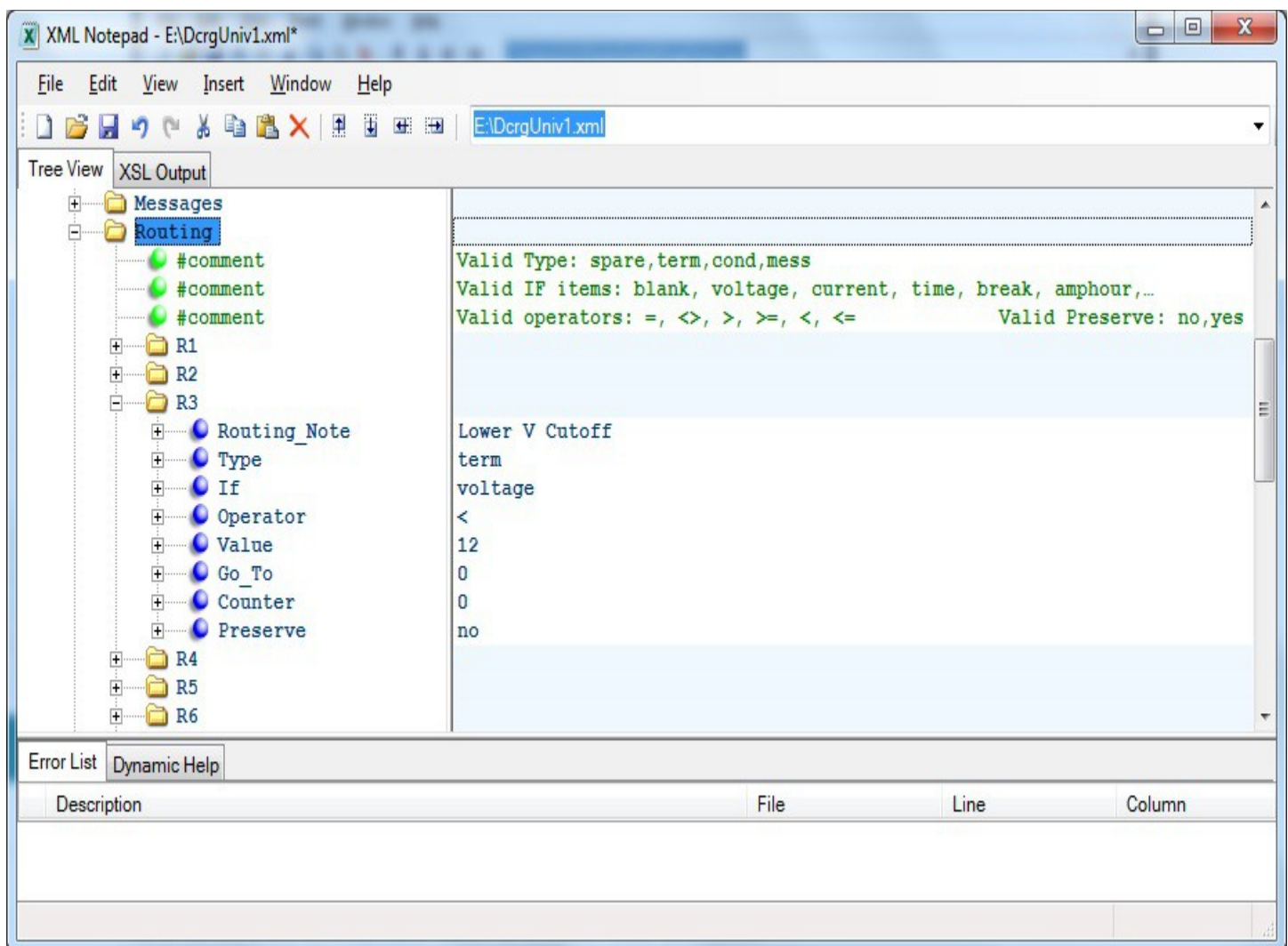
If a desired message cannot be found in the predefined list, then use the custom message option. The four message lines in this section are for custom text messages. You have the option to specify up to 4 custom text messages. If you don't see the message you want in the Message Number Listing below, then enter a custom message on lines M1 to M4. These custom messages can be up to 16 characters long and are referenced in routing statements by using message assignment numbers 1 to 4. In this way the four custom messages defined in this section are used just like all of the other predefined user messages.

Message Number Listing

0 – Unused	40 – Closed	80 – Spare
1 – Custom1	41 – Done	81 – Spare
2 – Custom2	42 – Interrupted	82 – Vector Shut Off
3 – Custom3	43 – Waiting	83 – Press Restart
4 – Custom4	44 – Good	84 – Code 1
5 – Connect correct battery to begin	45 – Bad	85 – Code 2
6 – Remove Battery	46 – Unknown	86 – Code 3
7 – Charging	47 – Marginal	87 – Code 4
8 – Trickle Charging	48 – Excellent	88 – Code 5
9 – Float Charging	49 – Poor	89 – Max Output
10 – Complete	50 – Re-Test	90 – Under Voltage-Check Bat
11 – Charge Complete	51 – Report	91 – Over Voltage-Check Bat
12 – Discharging	52 – Inspect	92 – Charged
13 – Discharge Complete	53 – Trouble	93 – spare
14 – Cycles Complete	54 – Trouble 1	94 – Cycle Charging
15 – Ready to begin	55 – Trouble 2	95 – Cycle Discharging
16 – Standby	56 – Trouble 3	96 – Press Start/Vec to proceed
17 – Wait...	57 – Trouble 4	97 – Precharging
18 – Print Report	58 – Trouble 5	98 – Max Input
19 – Test Complete	59 – Pending	99 – Timeout
20 – Vector to start	60 – N/A	100 – Resting
21 – Reduce Current	61 – Error	101 – Temperature
22 – Spare	62 – Error 1	102 – Spare
23 – IR Test	63 – Error 2	103 – Spare
24 – Spare	64 – Error 3	104 – Spare
25 – Spare	65 – Error 4	105 – Spare
26 – Spare	66 – Error 5	106 – Spare
27 – Spare	67 – Trickle	107 – Spare
28 – Spare	68 – Float	108 – Spare
29 – Pass	69 – Cool	109 – Spare
30 – Fail	70 – Alarm 1	110 – Spare

31 – Ready	71 – Alarm 2	111 – Spare
32 – Running	72 – Alarm 3	112 – Spare
33 – Stopped	73 – Alarm 4	113 – Spare
34 – Caution	74 – Alarm 5	114 – Spare
35 – Check	75 – Equalize	115 – Spare
36 – Ok	76 – Safety Stop	116 – Spare
37 – Warning	77 – Check Cables	117 – Spare
38 – Working	78 – Battery Connected	118 – Spare
39 – Open	79 – Battery Disconnected	119 – Spare

7.4 Program Routing



Routing Statements are defined in this section. Routing statements are used to control the operation of the program. They specify how each step will terminate and what will happen when the termination occurs. Up to 32 Routing statements can be specified. Specifying a Routing Statement does not mean that it will be used. In order to use the Routing Statement it must be assigned to a particular step in the Steps section. Routing Statements can be assigned in more than 1 step. The Routing Statement defines a decision that the device will repeatedly examine and execute whenever it happens to be true. There are 3 types of Routing Statements:

1. **Termination:** Used to generate a step termination. All termination statements are examined by the micro controller approximately once per second. When a termination statement is true the step will terminate. This same termination statement will also specify the next step the routine should execute. It will also specify if a counter should be incremented and if the accumulated data should be preserved.
2. **Conditional:** Conditional routing statements are only used by the routine when a step terminates based on a termination statement. Once a termination has occurred then all conditional statements assigned to the step are also examined. The first conditional statement found to be true will over-ride the *GoTo*, *Counter* and *Preserve* values specified in the termination statement that caused the termination. The termination will still occur, but these other items that specify the action to take are changed. Conditional statements are examined from the lowest statement number to the highest. The first conditional statement found to be true is the one that takes effect.
3. **Message:** Message routing statements specify which text message will be displayed in the console *Results Data* area. A message is very useful because it can examine the numerical data from the terminating and provide a text answer to the results. For example you can have the message statement calculate if a parameter is within range or defined threshold and give the user a meaningful message such: Pass, Fail, Good, Bad, Marginal etc. etc. There is a wide range of messages to select from. See *Program Messages* section for detail.

When the Program listing is Opened & Previewed in Battery Console, the format for Routing Statements will look like:

Rn:(type) If parameter operator value GoTo/Mess step (comments)

- **n** – Routing Statement number
- **type** – Defines type of routine statement. Must be: spare, term, cond, mess
- **parameter** – One of the available operational parameters. See list.
- **operator** – Operator used to determine if routing statement is true/false. See list.
- **value** – numerical value the parameter is compared to
- **step** – For term & cond types this is the step number the program will route to when statement is true. For mess types this is the message number.
- **preserve** – This optional item will appear in the preview listing only if *preserve=yes* otherwise *preserve=no* is assumed.
- **Inc CountX** – This optional item will appear if a Counter is assigned to this routing statement.

Example: R8:(term)If voltage < .75 GoTo 2 (Battery removed during session)

Build these routing statements by specifying items in this *Routing* section of the xml file.

Routing_ This is just any helpful comment used to document the statement.
Note

Select from: **spare, term, cond, mess**

Type

- **spare** – Routing Statement is unused
- **term** – This Routing Statement will be used to generate a termination
- **cond** – This Routing Statement will only be examined when a termination occurs. If this cond Routing Statement happens to be true then this GoTo step number will determine the next step. This cond Routing statement GoTo step number will over-ride the term GoTo step

number. Note that cond statements will not generate a termination. They only provide for alternate routing if and when and term statement is true. If more than one cond statement is true then the first one evaluated (lowest number) will be the one that take precedence.

- **mess** – This statement is only used to calculate a message for display purposes whenever a termination occurs.

This is the **parameter** that gets compared to the specified **value** using the specified **operator**. Use all lower case and observe proper syntax. Select from:

- **blank**: Routing Statement not used.
- **voltage**: Battery voltage(V).
- **current**: Positive value expressed in Amps(A).
- **time**: Step Time expressed in minutes. Minimum Value 0.02min(1.2sec), Maximum Value 938249min(651 days). Do not use Value=0 because it will disable the term statement.
- **break**: Special time counter used to break away from a step, perform some function (eg. IR test) and return to step keeping Step Time and other accumulated values intact. Use Preserve=yes when breaking and Preserve=yes when returning.
- **amphour**: Positive accumulated charge value expressed in AH.
- **watthour**: Positive accumulated energy value expressed in WH.
- **%capacity**: *Measured AH/Rated Capacity* expressed as a percentage.
- **%watthour**: *Measured WH/Rated WH* expressed as a percentage.
- **tapercurrent**: *Measured Current/Rated Capacity* expressed as a percentage.
- If**
 - **negdv**: This is how much the voltage has dropped from the peak charge voltage. Used to terminate nicd.
 - **temp**: Battery Temperature(°C) as measured with the thermistor cable.
 - **dttdt**: Rate of temperature rise(°C/Min) as measured with the thermistor cable.
 - **irtest**: Battery Internal Resistance(mΩ).
 - **chargefactor**: Charge Factor is the ratio of the mAH for this step to the mAH for the last Discharge step since a reset. The device will remember the mAH from the last discharge and calculate the Charge Factor on subsequent steps. This can be used to charge as a % of a previous discharge, or perhaps even perform another cycle based on the results of this step. This can save time by allowing you to do conditional charging, discharge or cycling based on the results and not have to perform more steps than required.
 - **bulktimefactor**: When termination is based on current, the device will remember the step time and calculate a Bulk Time Factor on subsequent steps. The Bulk Time Factor is the ratio of the Step Time from the last step, that terminated on current, to this Step Time. This allows for equalization charging and other enhanced charging algorithms.
 - **counter1-7**: Route based on the value of counter1 to counter7.

Operator Select from: =, <, >, >=, <=

Value Numerical value that parameter is compared to. Note: A value of 0 will disable the term statement.

Go_To For **term** or **cond** statement types this is the next step that will execute when this step terminates. For **mess** statements this is the message number to be associated and displayed with the Results Data. Using Go_To = 0 will mean advance to the next step. This is very useful because it means the termination statement is generic and can be used many times in multiple steps. This will reduce the

number of routing statements required because you can use the same one in many different steps. Statements assigned in multiple steps must be of the same *Type*

There are 7 internal counters available. They can be used for information or to control advanced program routing. Putting a counter number from 1 to 7 will cause the counter to increment when the step terminates. **Note:** when a step is terminated the *Results Data* is saved to memory before the counters are incremented. This means when using Counter 1 as a cycle counter you have to increment Counter 1 before beginning the step that will eventually correspond to Counter 1. This way Counter 1 (#) in *Results Data* will correspond correctly to the right step. This may mean inserting an extra step for the sole purpose of getting the counter to increment before a new step begins.

Counter

- **Counter 1:** This is normally used as a cycle counter and is displayed in the Real Time Data Cycle: field. It is displayed in Saved Data in the " #" column. This counter is cleared at the beginning of each Reset step
- **Counter 2:** General use counter displayed in Real Time Data section. This counter is cleared at the beginning of each Reset step
- **Counter 3:** A special counter that only gets reset on power off/on. It does not get cleared with a new session. It can be used as a session counter for daily/weekly production counts. This counter is incremented at the beginning of each Reset step.
- **Counter 4:** A special permanent counter that gets saved to flash memory. This counter can be used to keep a record of production. Like a car odometer it is never reset. Count 4 is displayed in the lower status bar section 8 on the main window. This counter is incremented at the beginning of each Reset step.
- **Counter 5:** General use counter. This counter is cleared at the beginning of each Reset step
- **Counter 6:** General use counter. This counter is cleared at the beginning of each Reset step
- **Counter 7:** General use counter. This counter is cleared at the beginning of each Reset step

In some cases it may be desirable to keep accumulated values, such as mAH, StepTime, etc. when advancing to another step. This allows the new step to just continue on. Normally you would always clear these values. However the data can be preserved when using an algorithm that is still continuing the same task with different parameters. Use 1 of 2 possible values:

Preserve

- **no:** Do not preserve any real time data when advancing to the next step.
- **yes:** Preserve accumulated data when advancing to the next step.

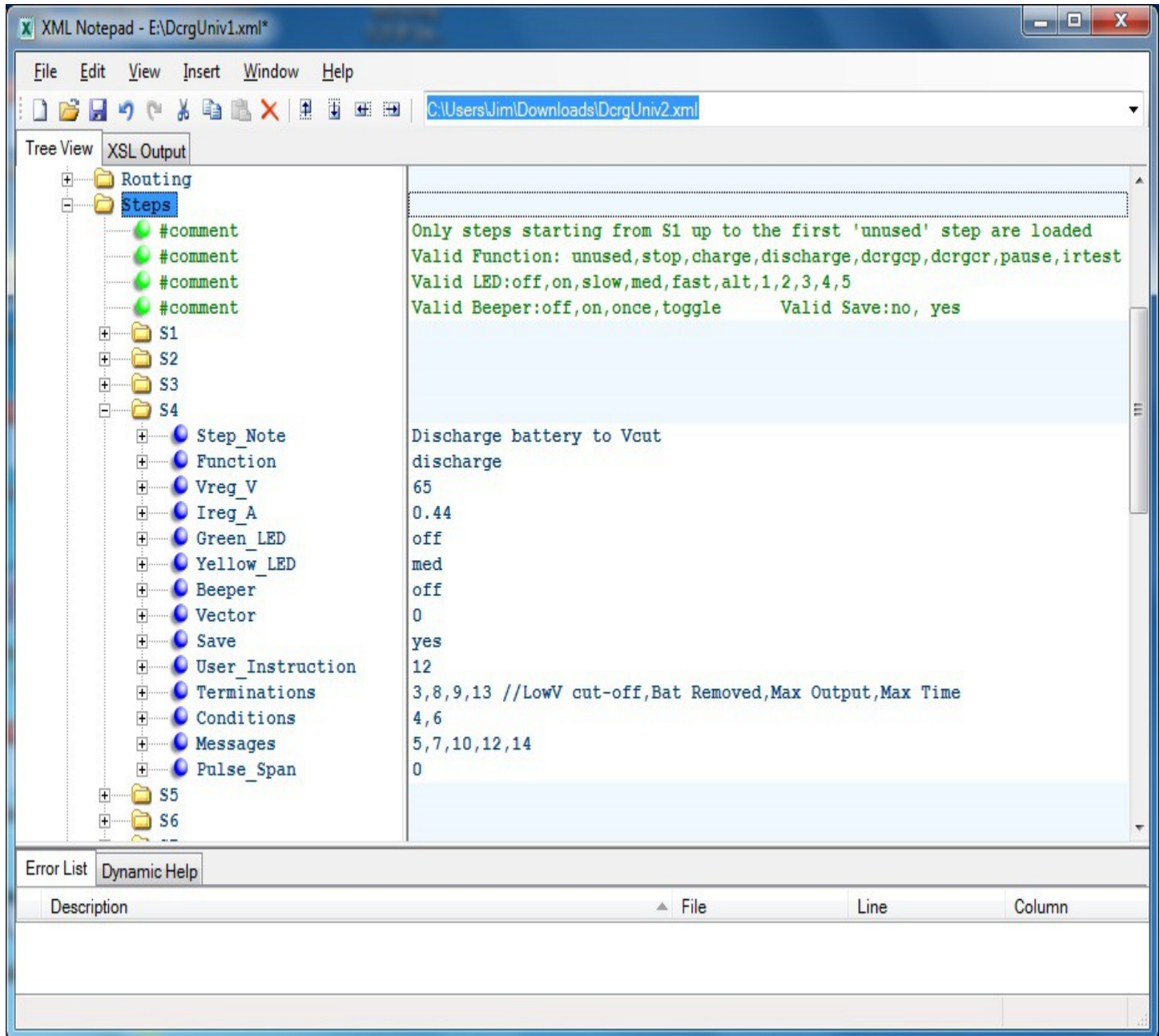
Routing Notes:

- Any step that does not have a method of termination defined will never advance to another step, unless, of course, it is forced with a vector command such as the software "Restart" or the hardware "Vector" push button.
- Any Routing statement that is not assigned in some step is simply ignored. Leaving unused Routing Statements in the listing for possible future use is fine.
- When preparing custom routines be aware that you must Save the data for any step that is to appear in the *Results Data*. Do this by setting Save=yes for the desired steps in the xml program file. Also note that Counter 1 is a special counter that gets saved with the Results Data and is used to assign a number(#) to the Result Data row. *Results Data* is saved to memory before counters are incremented. Therefore it is necessary to increment Counter1 before beginning the step that will eventually correspond to Counter 1. Keep this in mind when programming custom routines to ensure that

Counter1(#) will match with the correct *Results Data* row.

- If a statement is assigned in more that one step, it must always be assigned as the same type: term, cond, mess.

7.5 Program Steps



Step_Note This is just any helpful comment used to document the step

Define one of the following Step Functions:

Function

- **unused** – This means the step is not used and all subsequent steps following the first unused step are ignored and will not be loaded into device.

- **stop** – This step is the same as Pause step however the Step Time and Total Time clocks will not increment. Use Stop steps to wait for certain conditions to be met. They can also be used as interim steps to examine data for conditional routing. Use termination statements to advance to other steps.
- **charge** – Charge battery with maximum current limited to Ireg and maximum voltage limited to Vreg. Specify Vreg_V and Ireg_A.
- **discharge** – Apply a constant current discharge to the battery. Specify Vreg_V and Ireg_A when using the discharge function.
- **dcrgcp** – Discharge battery at constant power.
- **dcrgcr** – Discharge battery by applying a constant resistance load.
- **pause** – This will stop all charge and discharge current. Use Pause steps to wait for certain conditions to be met. They can also be used as interim steps to examine data for conditional routing. Use termination statements to advance to other steps.
- **irtest** – Use to perform a Internal Resistance test. This is a two pulse test where IR is calculated as: $R = \Delta E / \Delta I$. The result is reported in $m\Omega$. Note that an irtest will still need a termination statement to terminate the step and specify the next step. The IR test only takes a few mS so step termination could be based on step time. eg. if steptime>0.1 GoTo x

Vreg_V	The upper voltage regulation limit. The voltage will be regulated at or below this value during charge . This is the Voltage regulation value used for Constant Voltage(CV) charging. When managing batteries that only require constant current (eg. NiMH,NiCD,Primary etc.) set Vreg = 65(max). For discharge where Vreg is not required set Vreg = 65(max). No current will flow if Vreg is set below actual battery voltage.
Ireg_A	The maximum charge current allowed. This is the charge current value for constant current charging. This is also the initial current when charging batteries that require voltage regulation (eg. LiIon,LiPO, SLA etc.)
Green_LED	Specify the desired Green LED action for this step. Each step can have it's own unique led operation. Select from: off, on, slow, med fast, alt, 1, 2, 3, 4, 5.
Yellow_LED	Specify the desired Yellow LED action for this step. Each step can have it's own unique led operation. Select from: off, on, slow, med fast, alt, 1, 2, 3, 4, 5.
Beeper	Specify the desired Beeper sound for this step. The audio indicator can inform the user that tasks are complete or to sound an alarm as the situation may dictate. Select from:off, on, once, toggle.
Vector	Specify a step number to over-ride the default Vector setting. Vector is the step the routine will route to when the on screen <i>Start</i> button or hardware <i>Vector</i> push button is used. This provides for control and flexibility for the user to force the routine to move to another step as the application requires. If this value is 0 then the default Vector, as specified in <i>Details</i> , applies to this step.
Save	Specify if the routine should save <i>Results Data</i> for this step. Data that is saved will appear in the <i>Results Data</i> section of the main screen when the step terminates. Not all steps require data to be saved for display. Only steps where actual battery management activity is occurring will need to be saved. Many steps are just used for routing, control, messaging or to manage the LED & Beeper indicators.
User_Instruction	Specify a text message to instruct the user or inform them about what is happening. Each step can have it's own unique user message. Enter the number of the text message required. See <i>Program Messages</i> section for the list of standard messages or specify 1 of the 4 custom

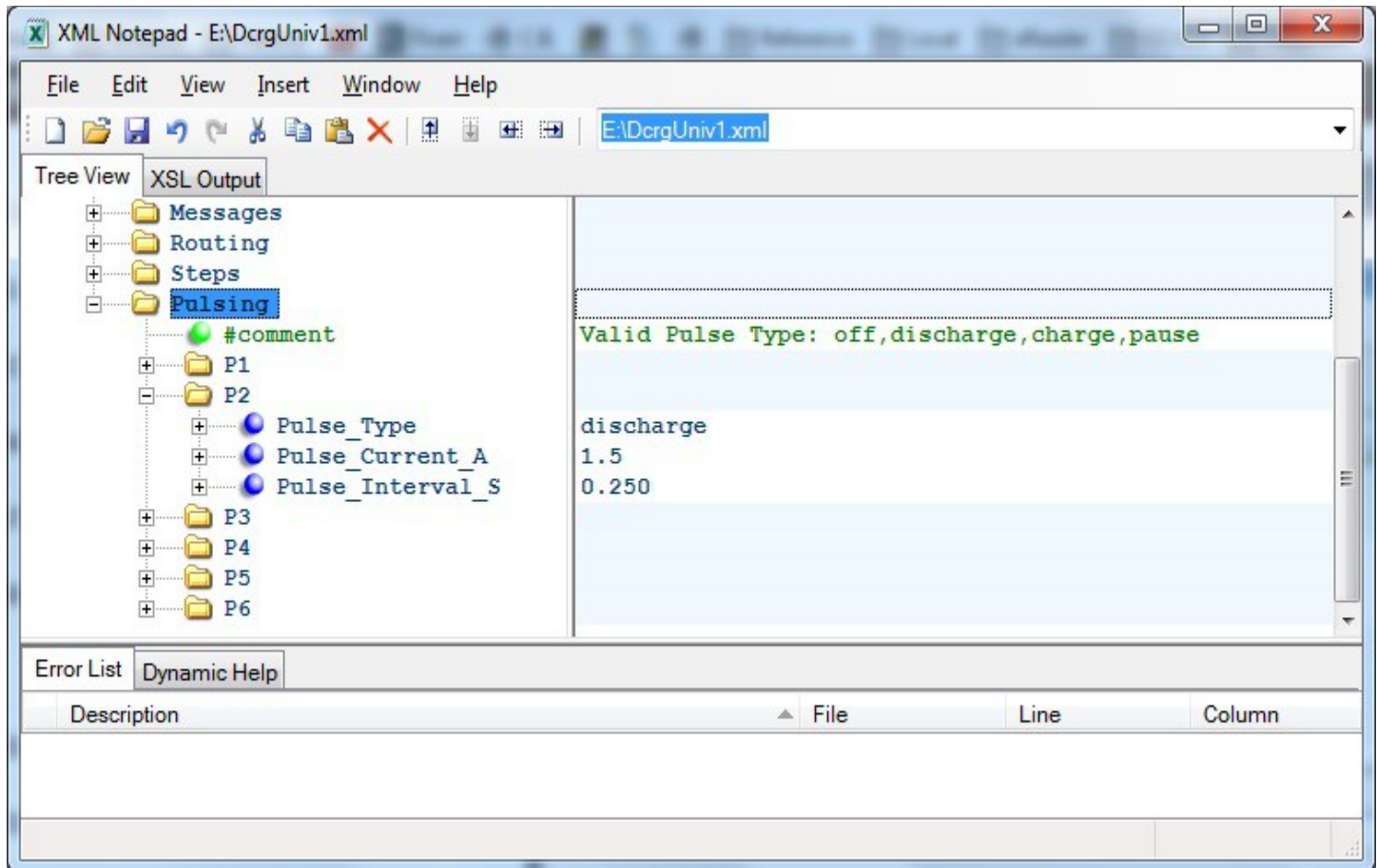
message numbers. The user messages appear in Section 3 on the main screen, while the step is in progress.

Terminations	Assign the routing statement number(s) for all statements that should be used as <u>terminations(term)</u> on this step. Separate numbers with comma. (ex. 3,8,12)
Conditions	Assign the routing statement number(s) for all statements that should be used for <u>conditional(cond)</u> routing on this step. Separate numbers with comma. (ex. 5,6,14)
Messages	Assign the routing statement number(s) for all statements that should be used for <u>message(mess)</u> calculation on this step. Separate numbers with comma. (ex. 2,9,13)
Pulse_Span	Enter 0 when not assigning pulses. To assign pulses enter start and ending pulse number separated by a comma (ex. 3,5) to turn on pulses 3 to 5 repeatedly. Pulses are defined in the <i>Pulsing</i> section of the xml file.

Additional Step Notes:

- Only steps up to the first *unused* step are loaded into the device.
- Interim Steps reserved for future use during development can just be assigned a function stop or pause.
- To make LED's alternate, set Green_LED=alt and set Yellow_LED= slow, med or fast to control the rate of alternation.
- To make LED's flash simultaneously, set both LED's to the same frequency. (ex. slow, med, fast).
- The LED's together with the audio Beeper allow for many, many different output combinations to indicate:
status, operation, warnings, errors, alarms, pass, fail, marginal, complete ready, charged, discharged
 or whatever the situation requires. These indicators are especially useful when using the device in stand alone mode. In a production environment they can prompt the operator to take action as required. Once programmed, the device does not need a PC connection so it can be used in stand alone mode. This allows the device to be portable and used in a wide variety of applications.

7.6 Program Pulsing



Pulsing allows for fast current changes with a step. Use this section to define up to 6 pulse definitions. These pulse definitions are then activated by assigning them in the desired step *Pulse_Span* field. See *Program Steps* section. Pulses are assign in the *Pulse_Span* field by specifying the first and last definition to be used. Of course at least two definitions will be required in a *Pulse_Span* in order to establish the pulse train. For example: *Pulse_Span* = 3,4 would create a pulse train consisting of pulse 3-pulse4-pulse3-pulse4... continued repeatedly until the step terminates.

Pulse_Type off, discharge, charge, pause

Pulse_Current_A Specify desired pulse current value in Amps

Pulse_Duration_S Specify length of pulse in Seconds. Maximum 6870 seconds

If you want to apply pulsing within a step then define the pulses here and assign a start and end pulse number as *Pulse_Span* in the step details. When the step begins the pulses will begin starting with the first number and proceeds sequentially up to the end number. This pattern then repeats indefinitely until the step terminates.

There are 3 different pulse types to choose from:

1. Discharge: Applies the specified discharge current for the specified interval.
2. Charge: Applies the specified charge current for the specified interval.
3. Pause: Applies a pause (Current=0) for the time interval specified. The current value is ignored.

Set pulse type=off: when pulse is unassigned and unused

Pulsing is often used to simulate the load that a battery may experience in the target application. By using this pulsing feature you can have pulses of very short duration. Pulses as short at 100µS (0.0001 sec) can be specified here.

Pulsing could also be created by cycling through a series of individual steps however the minimum time for an individual step is approximately 1.2 sec.

When looking for longer duration pulsing, pulsing patterns can be created using a series of steps in the routine. In which case you don't need to use these pulse definitions. These pulse definitions are really only required for higher speed pulsing within a single step.

7.7 Program Strategies

This section outlines some of the issues to consider when programming or editing your own battery routines.

Programming instructions can be used as in as many steps as required. The same programming statement can be assigned any number of times. They are especially generic if they do not specify a GoTo = 0 (ie next step).

Whenever the GoTo Step is 0 it instructs the device to route to the next numbered step. This is very common and allows you to assign the same program instruction in many different steps. With this strategy you will need fewer program statements. Also the program will be easier to read, understand, modify and debug.

Termination Program Statements

A termination statement is a routing statement that describes a condition which will cause the presently running program step to end. The termination statement consists of the condition to test and what action to take when the tested condition is met. Each step is assigned up to 12 possible termination statements. The termination statements are selected from up to 32 different programming statements available for each program. With each step you specify which programming statements to use for step termination. As long as no assigned Conditional Statements are true (see below) then the program routes to the step identified in the assigned Termination statement that caused the step to terminate. The assigned termination statements can also increment one of the 7 counters to when this termination statement is executed by the device.

Conditional Program Statements

Users can also specify Conditional statements with each step. This is optional and provides for alternate routing if specific conditions are met. With this feature the routing can branch to different steps depending on various parameters. If no conditional statements are specified then routing will follow to the GoTo step specified in the termination step that caused the termination. If conditional statements are specified then routing will continue based on the conditional statement if it is true. Conditional statements are optional. They are not required but they can be used to provide for more complex program routing that will be based on measured parameters. These are useful for writing programs that stop when some conditions exist or possibly go on to execute more steps or loops when other conditions exist. This can allow you to have the device routine finish sooner by running for only as long as required.

For example if the battery test runs successfully then the program can stop early to save time, however if more cycles or steps are required then the program will only run longer when required. Conditional statements are only checked once a termination has occurred. This is different than the Termination statements which are continually checked during the step. So once any termination has occurred all conditional statements (if any) will be checked. If any conditional statements is true then the Next Step, Counter and Reset parameters of the true conditional statement will override the Next Step, Counter and Clear parameters in the termination statement that generated the termination. The

termination will still occur as per the termination statement but routing will be controlled by the true conditional statement rather than the termination statement.

If there are more than 1 conditional statement then only the conditional statement that is true will be executed. If there are other additional conditional statements that also happen to be true then the first (lowest number – irregardless of order) true conditional statement will take priority. Once a termination statement terminates the step, all listed conditional statements are examined. Starting with the lowest number and continuing with the next higher number until all conditional statements are examined. If a conditional statement with a higher number is not true (false) then it will be ignored. If no conditional statements are true then the termination statement that generate the termination will control the routing.

By having the first true conditional statement become the one that controls the routing you can create very powerful selection algorithms to create selective routing. This can be used to implement a look up table. For example if you want to route to different steps depending on the voltage then you can write the conditional statements to look like:

- (1) If voltage <11.0 Go To Step 16
- (2) If voltage < 11.2 Go To Step 18
- (3) If Voltage < 11.4 Go To Step 20
- (4) If Voltage < 11.6 Go To Step 22 (Notice that if V=11.55 then the routine routes to step 22)
- (5) If voltage < 11.8 Go To Step 24
- (6) If voltage < 12.0 Go To Step 26

Note that this is not the same as:

- (1) If voltage <12.0 Go To Step 26 (Notice that if V=11.55 then the routine routes to step 26)
- (2) If voltage < 11.8 Go To Step 24
- (3) If Voltage < 11.6 Go To Step 22
- (4) If Voltage < 11.4 Go To Step 20
- (5) If voltage < 11.2 Go To Step 18
- (6) If voltage < 11.0 Go To Step 16

Because the FIRST true statement is the one that controls the routing.

You must have a termination statements for every step if you want to allow the routine to move to another step. If you do not specify any termination statements then the only way to force the device to route to another step is by doing a power On/Off (routes to step 1) or pressing the vector switch (routes to the vector step if it is > 0)

You do not have to have a Stop function in your program. Any step with no termination program statements will not continue anywhere and will act like a Stop function. The Stop function is similar to the Pause function with the only exception being that Step Time does not increment during a Stop function and it does during a Pause function. This means that you cannot terminate a Stop function with a program statement like: If Step Time > 0 then GoTo 6

However you could terminate the Stop function with a statement like:

If V<1 then GoTo 3, or If Step Time = 0 then GoTo 8

Hardware Restart

You can always generate a H/W restart by powering Off/On with the power switch. This is a hardware

restart. When the power comes back on the device will restart in Step 1. This method of restarting the device for a new session is really only needed when you cannot, or do not want, to implement an intelligent Software Restart (see below) to begin a new session. Normally you would use a software restart to have the final step check for a large voltage drop (battery is removed) and automatically proceeding to Step 1 to get ready for another battery to be connected. Step 1 would be programmed to check for a voltage rise (new battery is connected) and begin the session. However if you were using the device to test for an open battery situation then you may not want the open battery (no terminal voltage because battery is defective or battery protection circuitry has been activated) to restart the session.

Software Restart

In many applications you can write the program to automatically restart the program intelligently. This means you would implement a restart based on some event occurring. For example you could use a software restart to have the final step check for a large voltage drop (battery is removed) and automatically proceeding to Step 1 to be ready for another battery to be connected. Step 1 would be programmed to check for a voltage rise (new battery is connected) and begin the session. With this type of programming you never have to do a hardware restart. You can just keep reusing the device over and over again running the same routine again and again without ever having to power Off/On.

Counters

The 7 counters are useful as a memory device or to flag a particular condition. When a condition exists you can set a counter as a flag and then refer to this flag later.

ex. In Step 6 you may have: If mAH>1000 then go to step 14 and increment counter2.

Now in future steps you can just check counter 2 to determine if there was a Pass or Fail on this earlier step.

Counter 1: This is a special counter. The value of this counter appears in column 1 of the 'Saved Data' results in the console. Use this counter to count events and make sense of your 'Saved Data'. For example increment Counter 1 on completion of each cycle so that the cycles are numbered properly in the Console. Or if you are running a series of tests, increment Counter1 on completion of each test so the tests will all be conveniently numbered in the console 'Saved Data' area. This counter is also displayed as real time data in the console so that you can see which cycle number or test the device is currently processing..

Counter 2: This is a normal counter available for general use. It gets cleared on the Reset step and is visible as real time data in the console.

Counter 3: This is a special counter. This counter does not get cleared on the Reset step. This counter will only get cleared when the device powers Off/On. So you can use this counter to measure productivity on a daily basis. Whenever you want to reset this counter simply power the device Off/On. This is useful for counting the total number of tests performed since the last time it was cleared. This counter is displayed as real time data in the console.

Counter 4: This is a special counter. This counter never gets cleared. Even a power Off/On will not clear this counter. This is a permanent counter used to measure productivity over a longer period of time. Think of this as the lifetime odometer. Of course, if you want to use it, you will need to program the routine's programming statements to increment this counter. This counter is displayed as real time data in the console.

Counter 5, 6, 7: These 3 counters are normal counters available for general use. They get cleared on the Reset step but are not visible as real time data in the console. Use these as required to control routing of the routine.

Session Reset

In the programming information you specify which program step causes a session reset. Once the program encounters the step designated as a reset step then all data and information from the previous session will be cleared. This data will no longer be available. This could be set up as step 1 so that every time you turn on the device you will clear the previous session's data. Alternatively, if you want to preserve the data for later viewing set the reset step to a higher number so that the new session won't begin until the battery is connected.

A session reset will clear the Total Time timer as well as Counters and clear all saved data from the previous session.

Vector

In the programming information you can specify a vector step. This is the step that the device will route to when the hardware push-button is pressed. With this feature you could actually have 2 programs loaded in the device and the vector button can be used to force the program to the 2nd program routine. The vector push switch can be set up to have many different uses. e.g. Restart, reset, continue, stop etc. etc. You just need to specify a step to route to when the vector push button is pressed. You can also route the the vector step by pressing the Vector button on the software console.

If you have configured this device to be part of a group of devices (Host channel plus all devices that share the same host channel) then routing to the vector step will route all devices in the group to their respective vector steps as well. In this fashion the vector step can be used to control all devices in the group. This would normally be used to be able to shut down or reset all devices in the group simultaneously.

Program Debugging & Testing Tips

When trying to test your new programs you can generate a termination using Step Time. This would be used just for testing and is useful for testing conditional statements. By adjusting the termination statement time interval you can simulate a variety of conditions to ensure that your program is routing as you expect. When you no longer need this test termination, just remove the statement number from the step.

Routine statements are assigned as either termination, conditional or message statements by listing the corresponding program statement number. Remember, if you do not have any termination statements listed, then the step will never end. Also remember that conditional statements are only checked once a termination occurs. When a termination occurs as a result of any termination statement then the conditional statements are checked.

When a conditional statement is checked and found to be true, the program will act on this statement and route accordingly. True conditional statements will be ignored and not be acted upon until a termination occurs due to a true termination statement. The conditional statement, found to be true, will override the GoTo, Counter and Clear items of the termination statement that caused the step to terminate.

It is possible to have logical errors in your program that cause the device to act in unexpected ways. For example if you have a term statement such as: *If Step Time < 65 then go to Step 0 (next step)*.

This statement will immediately be true at the beginning of a step causing it to terminate immediately. The same can be true for all statement types. You need to carefully consider the effect of each step and then test your program carefully.

You need to test your program routine thoroughly so you know exactly how it will perform in all situations. Failure to test your program can result in the device acting incorrectly. This can cause a

dangerous situation resulting in battery damage, fire, explosion personal injury and death.

In your program a single programming statement cannot be used as termination statement AND a conditional statement. If you ever need an identical termination and conditional statement then you will need to specify two separate identical programming statements and use one as a term statement and the other as a conditional statement. With 32 possible programming statements you will probably never need to use this many in one program. You do not need to assign all the programming statements. You can keep them in your program so that your program list can act as a library of available programming statements. This can make programming faster and easier to adjust.

If you have specified charge pulses during a discharge then you will need to enter a maximum Vreg value. If no charge pulses are specified then a Vreg entry is not required for a discharge function.

Counters only increment after the step has terminated and just before the next step begins. This means that if you increment a counter, it cannot be tested for the new value until the following step.

Inserting some extra steps can facilitate program building, testing and debugging. If you have some spare steps scattered throughout your program then it becomes easier to build steps without having to readjust the termination and conditional statement numbers of existing steps. You can just use the spare steps.

Be careful with the use of the equals (=) function

Avoid the use of the = function for terminating values that can change, such as Voltage, current, time, mAH etc because the value could be lower then greater and never be equal. For example if the voltage is dropping during a discharge do not attempt to terminate the discharge with a statement like: If Voltage = 10 because the voltage could drop from 10.01 to 9.99 in the time between the device checks for a possible termination. Resulting in the discharge continuing right down to 0. This is an example of a logical program error and an example of how all programs must be tested carefully before they can be relied upon.

The correct terminating statement to use would be: If Voltage < 10 then

Keep this in mind when terminating on other values as well.

The = function should really only be used when programming statements with counters.

Using dT/dt termination

Allow time for the battery to cool if you need dT/dt termination on two successive steps. You can insert a pause statement to allow the battery to cool and then terminate the step when $T < \text{Value}$. This way you can have a pause that lasts just long enough for the temperature to drop to normal. dT/dt will only detect a positive increase in the rate of temperature rise. If the temperature is dropping then the dT/dt value will be zero.

Consider what will happen when the battery is disconnected

If the battery is disconnected then the voltage that the device sees at the terminals may not be zero. If you have a step programmed to charge with Vreg=12V and Ireg=100mA. Then the device will generate either 12V or 100mA whichever is reached first. When the battery is disconnected, no current will flow and the device will regulate the terminals to 12V. So the device will be measuring 12V.

If you are trying to use a step that will detect when the battery is connected then you should use a Vreg value that is below the expected battery terminal voltage (you could use Vreg=0) and then terminate this step on a voltage>1V . The voltage can then only be >1 when the battery is detected.

This is also important to consider when trying to detect a disconnected battery at the end or during a session. When the battery is removed the current will stop, so the device will regulate on the Vreg

value if it is a Charge step. If you have been charging at constant current then the Vreg value must have been greater than the battery voltage. The device terminal voltage will then rise up to the Vreg value once the battery is removed. So to detect the disconnected battery you should terminate on the voltage being greater than some value. For example to charge a 6 Cell 7.2 V NiCd battery you could set the Ireg to 200mA (desired charge current) and set the Vreg value to 20V. Once the battery is removed the device terminal voltage will rise to 20V. The device can be program to terminate if the Voltage>18V

Similarly if you are charging a SLA battery at constant voltage of 14.7V with a Ireg = 250mA (Maximum current limit) then when the battery is disconnected the device terminal voltage will continue to be 14.7V. In this case you can program the step to terminate (and route to step 1 to start over) when the Current is < 15 mA or some low value that will only occur when a battery is removed. You can also use an extra step to restart the device if you think there could be a possibility that the current would drop to less than 15mA normally. In this case you could have the device go into an extra step when the current drops below 15mA. This extra step could be programmed with Ireg=0 and Vreg=5V. Now this extra step would not provide any charge current because the Vreg value is below the actual battery voltage. Once the battery is removed the device will regulate to 5V and you could terminate this step (detect a removed battery) if the voltage < 7. 5V is a better choice than using Vreg=0 because the voltage may never actual=0 due to normal offsets. For example if you set Vreg=0 then the device may measure 0.05 V or some small value so the statement "If V=0 then" will never be true.

So remember that during a Charge step the terminal voltage will rise to the Vreg value when the battery is removed.

To detect a battery removed situation during a charge you will need a statement such as:
IF Voltage > 10 then Go To next step (This has now detected that the battery has been removed)

Double testing

You may want to consider the effects of vibration and user actions on the operation of your program. For example: suppose you are performing a Lilon charge and want to terminate the charge when the current drops below a certain value. In the event the user touches, jiggles, shakes, vibrates or momentarily disconnects battery then the step may terminate. If you feel this could occur and you want to protect against this possibility you can do a double test of the current.

To do this, route to the next step when the current drops below a certain value. The next step will perform the same function and terminate on step time in 0.1 minutes and route back to the previous step (normal charge step). Include a conditional statement in this second step that will check the current when the step terminates. If the current is less than your minimum then you can assume that the battery has been removed on purpose, however if the current at the end of the 0.1 min is not less than the minimum then it was just a momentary drop in current and the program can then route back to the original normal charge step.

If you want to skip the clear then you cannot use step time to create delay with your double testing. In this case you could implement a counter situation. For example set use a termination statement such as: If V < 6 Go To next step. On the next step terminate If V < 6 as well but go back to a previous step & increment counter X. On 2nd step also have a conditional statement that if Counter X>10 then continue on to the 3rd step. This will create a 10 step loop to provide for the desired amount of delay for double testing.

Debugging

Often it is very helpful to route to steps that are just used for parameter checking and routing and are not required for any length of time. On some of these types of steps you may choose to terminate on

Step time (ex IF Step Time > 0.1 M). When writing and testing programs with these types of steps you might want to use longer step times while you are building and testing your program. This will allow time to see the step displayed on the PC Monitor to confirm that the step is actually occurring and operating as expected. Once your program is working as required you can then shorten the step times to your desired values.

A nice way to develop programs is to periodically review the complete program listing using *Program Preview*. This provides a comprehensive listing of the entire program and gives a bigger picture of the overall operation. You can view this online or send a copy to your printer for review. A printed copy will provide a hard copy of everything you have so far and give you a place to mark revisions and plan how to proceed further. A printed hard copy is a nice working document to help visualize the operation of the entire program and spot items that need further consideration.

Developing and testing your program will take a bit of time so proceed carefully with thought and planning. Expect to make changes and add new functionality to your program as you proceed. As you program and test you will discover new things that you need to consider and incorporate into your program. Developing a new program file is an iterative process of: Edit – Load – Test, Edit – Load – Test, Edit – Load – Test until you are completely happy with the performance in all possible situations.

Smart programming

There are several tricks you can do to perform some complex checking within your program. For example if you want to determine if the battery voltage is between 0.5V and 7.2 V you could have one step to test if $V \geq 0.5$. If yes route to next step. The next step is identical in function but has term statement if $V \leq 7.2$. This will only occur now if V is in the range of 0.5 – 7.2 because the previous step determined that it was not < 0.5 V.

If you want to route to one step if a parameter is > than a certain value and route to an alternate step if the same parameter is < than a certain value you can do this all in one step. To accomplish this, set the term statement to some value that will definitely occur. ex if step time > 0 min. Then have two conditional statements one for Parameter A \geq number X and the other for Parameter A < number Y. These two conditional statements can then route to different steps. Note in this case the GOTO part of the termination statement will never be used because one of the two conditional statements will always be used.

Alternatively you could use the fact that only the first (lowest number) routing statement that takes effect to create a powerful look-up table. Described earlier.

Remember the Battery's effect

The battery of course is connected to the device during use. The battery has a voltage and there will be current flow at various times during the operation of your routine. You need to think about what the voltage will be, at various times, and how this may or may not affect the operation of your program. Also the battery voltage takes a finite amount of time to rise and fall once charge current begins and ends. An example is when charging a SLA battery at a constant voltage. You may want to move to a new step once the current drops below a certain point. If the following step is also going to regulate the voltage at some lower level, the current will probably be 0 for some small period of time until the battery voltage naturally drifts downward to the point that current will again start to flow. So if this second step is also going to terminate on some minimum current then it will probably terminate immediately because initially the current will be 0. You will want to plan for this when you prepare your routine.

Also when using the temperature probe remember that the battery will take a finite amount of time to cool. An easy way to avoid situations such as this is to insert a pause step as required that will give

the battery time to stabilize before proceeding to the next step.

Step Time vs Total Time

Remember that Total Time is the time elapsed for all steps since the last session reset. Don't confuse this with Step Time which is the time elapsed for each step.

Consider the User

Remember that the person using the device will not be thinking about how the program is structured. They may remove or reconnect the battery at any time. You need to consider this when writing your program. If you want the device to revert back to the start of the program when the battery is removed you need to think about how this will work for all steps. This can be easily accomplished with a bit of planning. Often the program will just proceed through several steps and reach back at the start again. It is possible however that the program could get stalled on a step if you don't consider all possibilities. If the device does get stalled on a step during program execution you can always power it on/off (hardware restart) to restart the device.

The device programming functionality is very powerful. With a bit of thought, planning and testing you will be able to write a routine to accomplish your objectives that is robust, easy to use and reliable. Don't be afraid to experiment, test, debug and revise until you are happy with the operation. Of course you must watch your device very carefully during program development to avoid dangerous situations of battery over charge, or over discharge. The battery can be damaged, catch fire or even explode if not charged properly. The best way to avoid this is to use a voltmeter and current meter to carefully monitor the voltage and current during program development. Monitor the Real Time and Save Data fields to monitor how the device is operating.

Only when you have fully tested your Program routine should you use it, or make it available for other people to use. Also each time you reload the Program you should monitor the process to ensure that it has been loaded correctly and is working as expected. Also be sure that you have indeed loaded the Program into the target device and have not inadvertently loaded it into another channel.

Mandatory Requirements

Companies, personnel and individuals who use this Battery Console software to load programs into the Battery Metric device should label the device correctly and accurately with sufficient detail so that people using the device will be sufficiently notified as to the specific application and use for the device. Just like every battery charger is labelled for the battery type, voltage, rating etc. you should label this device with similar information. This should include battery type, battery voltage and any specific instructions or warnings that the user should be made aware.

Furthermore, companies, personnel and individuals who run this Battery Console software must agree to sufficiently test the device after it has been programmed to ensure that the programmed device operates correctly for the intended application. This includes running each and every device with the Program prior to delivery to a user. This run test should include monitoring of voltage, current and all termination parameters for all steps of the program. The voltage and current should be monitored and measured using suitable current and voltage meters. Every possible situation that a user could encounter should be planned, accommodated and tested before putting the device into service.

Miscellaneous Notes

Note that the routine routes automatically to step 1 when the device is powered up. However the previous session's data does not get cleared until the routine routes to the reset step declared in the *Program Information*. This gives you the opportunity to use the device to test a battery. Then remove the battery, turn off the device and preserve the session details and data to be viewed at a different

time or location. For example you could test a battery at one location, then return to your office or shop to view & print the console information. This allows you to use the device as a battery tester in a remote location without requiring a computer. To take advantage of this feature you should write the routine to not reset until a battery is connected. This way you can turn on the device, view the on screen console details of the previous session on your computer. When you are ready to start a new session simply connect the battery. Ensure the routine is programmed to detect the battery voltage and route to the Reset Step thereby clearing all the data from the previous session and then the program can proceed with the new session.

Consider possible power failures

If the power fails during a session the device will automatically shut down all charge and discharge current. This is a fail safe mechanism designed to protect the battery from any damage due to over charge or over discharge. When the power is restored the device will automatically start at step 1 again. If power failures are a concern to you, then you can write the routine to perform as required during this situation. For example you could just have the session restart from the beginning. You could also have the device halt further action until you intervene. For example have the routine require the battery be disconnected at power up before proceeding to the next step. You can also preserve the console details until you get a chance to view the session up until the time of the power outage.

The following tips can help you set this up to meet your needs:

- Have step 1 only proceed if the battery is removed (i.e. $V < 1\text{V}$). You can view the console during this step.
- Use step 2 to detect if a battery is connected. So now you will have to remove the battery (step 1) and reconnect (step 2) to begin a new session.
- Use step 3 to reset the previous session's data so the new session can now begin.

7.8 Memory Organization

The Battery Metric device has 64 pages of flash memory available for your program. There are 3 requirements for the use of these memory pages:

1. Store the instructions for each individual step.
2. Save the results data after a step terminates for display in the virtual console.
3. Hold the memo data.

Each step will require one page to store the instructions for that step. The page number is the same as the step number. So, for example, if your program has 20 steps, then memory pages 1 to 20 will be used to store the step instructions. In this case the *Results Data* will be stored starting with Page 21 and will advance forward for each step that is programmed to save *Results Data*.

The memo data is a text listing that can be used to describe and document the operation of the program. It is downloaded to the device with the other program details. When the device is loaded with a new program the memo will be saved into the memory pages. It is loaded in reverse order starting at page 64 and proceeding forward until the entire memo is saved. Each memory page can store 32 memo characters so you can store quite a detailed memo. However if the memo is too long, it

will get cut off at the end when the program instructions are loaded. Also the memo may get overwritten during operation of the device if the results data requires more space than the available spare pages. If this happens it's not really a problem. It just means you should use a shorter memo.

When programming, only save *Results Data* for steps that you want to record in the *Results Data* section. This is good practice. Do not save any more data than required. Also unnecessary *Results Data* will clutter up the console *Results Data* display.

In summary, the 64 pages of available memory program are shared between these 3 uses. If you use less for one use then it is available for the other uses.

There is other data that is also stored in the device such as the routing statements, calibration data, battery details, program name etc. etc. However these other items have their own separate memory and are not competing for any of the 64 pages.

8 Current Sharing

The Current Sharing feature allows multiple devices to operate as one group to increase the current carrying capability. When operating as a group, one channel will be the designated host channel. The host channel is the channel that controls the operation, aggregates the data and displays data for the whole group. To set up a current sharing group in *Profile Setup* mode:

- Specify the *Group Size* for the channel that will be the host in profile *Options*.
- The *Group Size* should be a number of 2 or greater because a group will consist of 2 or more channels.
- The other channels in the group must be subsequently numbered channels. ie. if channel 5 is the host with *Group Size*=4, then channels 6,7,8 must be followers in the group.
- The host always has the lowest channel number.
- Each follower will apply the same charge and discharge current as the host channel.
- The charge and discharge current specified for the host channel will be evenly carried by each channel in the group.
- $\text{Current on each channel} = \text{Current specified in host channel} / \text{Group Size}$
- The host channel is used to manage the entire group. Specify *Profile Setup* parameters and control the group using the host channel.
- There can be more than one group operating at a time on other channels.
- Current, AH, WH data reported by all channels in the group is aggregated and displayed in the host channel.
- Charge and Discharge current in following channels will automatically start and stop in response to activity by the host channel
- Do not attempt to control the follower channels. All control is performed on the host channel.
- All channels in the group are connected to the same battery in parallel. Observe proper polarity.
- When operating a group, all channels in the group must be connected to the PC and be online. The *Battery Console* software application must be running because the Host to Follower commands pass through the PC.
- If *Battery Console* is closed, then all follower channels will shut down after 30 seconds for safety.
- A current follower always uses the high range.
- The current resolution for all followers is 0.01A
- Followers are only set up to charge and discharge at constant current(CC).
- Constant Power(CP) and Constant Resistance(CR) discharge are not available at this time.

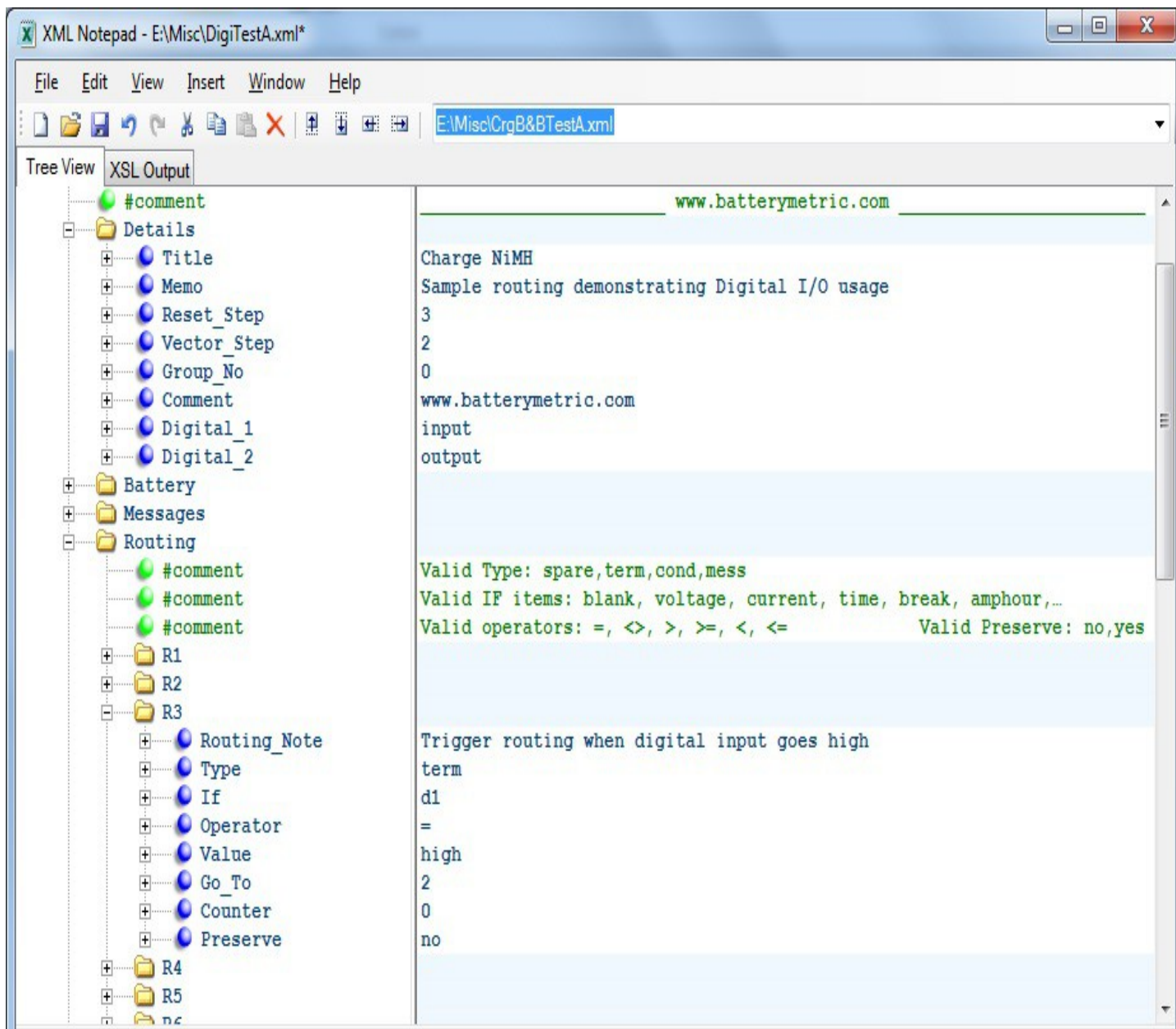
Current Sharing is not available in Quick Setup.

Current Sharing for custom Routines is enabled by specifying *Group_No* in program *Details*

9 Digital I/O Ports

Battery Metric devices feature two digital Input / Output ports D1 and D2. They can be configured as either inputs or outputs.

D1 and D2 can be used to have the device control external circuitry or have external circuitry trigger step terminations to route the programmed routine as required. The I/O ports are configured and setup up when the battery test routine is developed. This is done in the XML routine file. See Programming section for more details on editing XML routine files.



In the example above D1 is configured as an input and D2 is configured as an output. D1 and D2 can be used independently. Use both as inputs, both as outputs or one of each. Default values for D1 and D2 when not in use is input.

Now that D1 is configured as an input, it can be referenced in a routing statement. Specify d1 or d2 next to the *If* tag in the routing statement. The routing statement number then gets assigned to the desired step. Now when the assigned step is operating, the device will monitor the value on D1 and generate step terminations as defined.

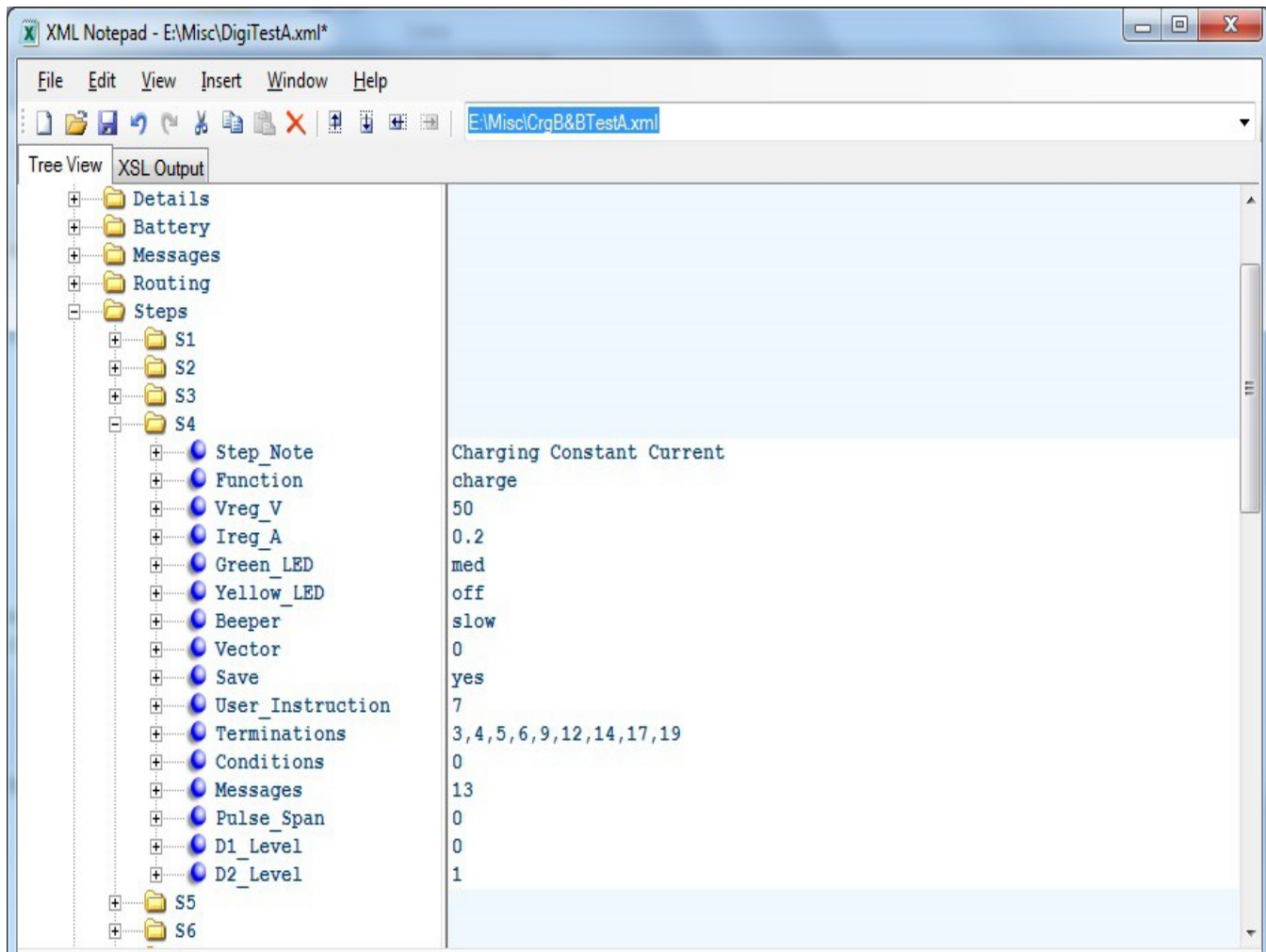
Since D2 is configured as an output it is assigned a digital value in the desired step next to the Dx_Level tag.

Port Direction	Usage
Digital_x = input (x=1 or 2)	D1 and D2 used to control routing. Use by creating a routing statement with value set to “low” or “high”. Then assign the statement number as a term in the appropriate step. External equipment can now trigger step terminations to control the program routine.
Digital_x = output (x=1 or 2)	When D1 and/or D2 are setup up as outputs they can be used to provide digital signals to external equipment. Specify Dx_Level = 0 to output current sinking ground potential of 0V. Specify Dx_Level = 1 to output current sourcing potential of 5V.

Do not apply external voltages when D1 or D2 are being used as outputs. When using D1 and D2 it is a good idea to use a series resistor (1K Ω) when connecting to the D1 and D2 port pins to avoid the potential of damaging the port in the event a conflict occurs. ex Connecting external +5V to a data port accidentally configured as an output.

D1 and D2 as outputs could be used to power LED's. Use a current limiting diode of 470 Ω . This is useful when developing or editing routines. Avoid applying voltages (high) to digital inputs when the Battery Metric device is off or during power on. Maximum current source or sink should be limited to less than 12mA.

Any assignment to the Dx_Level in a step will be ignored if the Dx port is being used as an input.



Syntax for assignments must exactly match the examples shown in these two screenshots. This is true for both the XML tags and the assigned values. If the XML file does not have the appropriate digital tags then just create them in the XML editor program observing proper syntax.

Digital inputs are expressed as either high or low in the routing statements next to the value tag. Digital outputs on the other hand are distinguished by 0 or 1 in the desired step parameter next to the Dx_Level tag.

Optional smart battery fixtures are available that will fix certain charge and discharge parameters with hardware. This is to allow a universal routine to operate with different batteries or prevent users from changing critical operating parameters. RatedCap, Vreg, CrgReg, DcrgReg, Vcut, are then fixed in hardware. D1 and D2 are used to communicate with smart battery fixtures. In this case set:
 Digital_1 = sba, Digital_2 = sba
 Contact Battery Metric for more information on smart battery fixtures.

10 Calibration

The MC, LC, RM and MK devices from Battery Metric are factory calibrated. The calibration data is stored in the flash memory of the device and all calibration parameters are automatically applied to all regulation and measurement functions. User calibration tools are available in the BA500WIN software application should users wish to perform recalibration procedures. See the BA500WIN manual for more information.

Upgrading the firmware will not impact the calibration settings in anyway. Calibration data is kept in a separate memory area and is unaffected by a firmware change.

Calibration services are available from Battery Metric. For more information visit:

www.batterymetric.com/calibration

11 Application Ideas

Take advantage of various device resources in order to use the device in innovative ways. The following examples give you some ideas. You will undoubtedly come up with other ideas of your own. The device has a unique design that allows you to program it in an unlimited number of ways for all sorts of applications.

Decision Making: For example you can use the device to make decisions based on measured parameters. A good example of this would be to create a single program to charge 6V or 12V SLA batteries. The device can detect the voltage and proceed to the step that will apply the correct charge parameters for the detected battery voltage. The green LED could illuminate for a 12V SLA and Yellow LED for a 6V SLA

Power Supply: The device can be used as a power supply. It can be set up as either a current supply or a current limited voltage supply. It is a very simple program that would provide the specified voltage / current at the output terminals.

Multi-staged Charging: The device can be programmed to charge in various stages. This could be a high current initial charge followed by another stage with lower current. Any number of stages can be programmed and rest periods can be inserted anywhere as required.

Function Generator: With the pulse capabilities the device can provide current or voltage pulses.

Battery Monitor with Alarm: The device can be programmed to monitor a battery voltage. If the voltage gets out of range the device can sound the built in alarm or flash a corresponding LED.

Load Module: With discharge capabilities the device can be programmed to be a constant current load module. This could also include options to interrupt the load if certain voltage, current or time parameters are reached.

Voltage Monitor: Simply use the device to monitor voltage. It could provide audio, visual or special messages when the voltage is not within specified thresholds.

Temperature Alarm: With the optional temperature input and temperature probe the device can be programmed to monitor temperature. If the temperature gets out of range the device can sound an alarm or flash a corresponding LED. This could be battery temperature, room temperature or any other temperature.

Float Charger: Hold an SLA battery to a float charge.

Field charger with 12V Input: Some devices can be powered by a 12VDC source instead of the external 24VDC supply. With this feature you can power your device with a 12V car battery. This could be used for turning your device into a charger you can use on the go. The maximum output

charge voltage will be about 8V when powering the device from a 12VDC supply.

Trickle Charger: Great for maintaining a NiCd battery in a ready state of charge

Cycler: Program any number of cycles

Conditioner: Exercise a battery

NiCd Refresher: Discharge to a lower voltage per cell to equalize the cells

Timer: The built in timer can be used to provide timed charges or even to just sound a beeper after an elapsed time.

Battery Life Cycle testing: The device can be programmed to simulate a real world application that a battery may experience. The device can be used to see how a battery will respond using this simulation. The simulation can include discharging and charging as well as using pulse currents to simulate changing loads. The LED's and beeper indicators can be programmed to let the user know when certain conditions are met. This can be an invaluable tool for a designer trying to evaluate various batteries and battery requirements.

Commissioning and battery preparation: Provide set amounts of charge to prepare batteries for shipment.

Your imagination is the limit. Think of the device as a battery charging & testing module with a built in operating system. This, together with the Battery Console software application, allows you to write your own battery programs. This puts you in control to build the ideal module for your unique application. Once you have created your routine you can load or modify it as required. A battery testing system that is not limited to what someone else designed. You can make it run as you need it.