TECHNICAL ADVICE 18-01

Retaining Setpoint Information for Schneider PLCs

1. Background

Setpoints for programs in PLCs can be changed from time to time due to tuning of the operating parameters. The changes may be made locally, via an OIP, or remotely by an operator.
However the setpoint information in Schneider PLCs may be cleared to zero when downloading a new configuration. This can lead to considerable restoration effort unless the setpoints have been backed up or written down.

This Application Note provides a process that, if followed, will prevent the loss of the variable setpoint information.

2. How to prevent loss of setpoint information.

Project phase
Enable the mechanism for updating initial values with current values by enabling the "Save" attribute.
In the Data editor you can display the "Save" column by right clicking on the column and selecting 'Customize Columns' which launches the "Column Configuration" tool.

On-site phase
Before any configuration change is made do the following:

- Unity Pro project must be online and equal to the PLC
- Use the command "Update Init values with Current values" from the PLC menu.
- Update the project using the command "Update Local Init Values with PLC Init Values" from the PLC menu.
- Save the project
- Verify in data editor that the variables with the save attribute have been updated

The appendices are extracts from the Schneider Electric FAQ site and provide some more detail and options.

3. Conclusion

Setpoints in a program for a Modicon PLC can be retained during configuration downloads if the correct steps are followed.
The steps, outlined above and below, should be followed as a matter of course for configuration downloads.

Enquiries regarding the above revisions can be directed to jim.baker@watercorporation.com.au or by phoning (08) 9420 3220

Appendix 1: Extract from Schneider Electric FAQ (FA197511)

How to replace the initial values with the current values in the PLC and in the Unity Pro project

Reminder :
During the development of the project you may define some initial values for the located variables (with addresses) and the unlocated variables.
After a download these initial values are in a specific zone within the PLC memory in order that after the cold start the initial values will replace the current values.

Based on process needs, sometimes the initial values in the PLC need to be replaced by the current values during PLC runtime. With a few steps, these current values can be made the initial values in the Unity Pro project. Below are details for setting the current values to initial values for both the project and CPU.

Data Editor :

To enable the mechanism for updating initial values with current values, the variable must have the "Save" attribute checked.
In the Data editor you can display the "Save" column by right clicking on the column and selecting 'Customize Columns' which launches the "Column Configuration" tool.

Name	T...	Address	Value	Save
Level_Product1	INT		100	☑
Speed	INT	%MW100	1	☑

1. **Replacing the INIT values with current values in the PLC (Run time) :**

    a. Method using logic.
    Setting system bit %S94 will move the current value to the INIT value column for all variables which have the SAVE attribute checked.
    Once this is complete, %S94 will reset itself, do not hold it on.

    b. Method using Unity Pro.
    The Unity Pro project must be online and equal to the PLC .
    Use the command "Update Init values with Current values" from the PLC menu.

2. **Replacing the INIT values with current values in the Unity-Pro project :**

    Using Unity Pro Version V6.0 (and greater)

    a. Replace the init values with current values in the PLC (Run time) using one of the two methods describe above.
    b. Update the project using the command "Update Local Init Values with PLC Init

Values " from the PLC menu.

c. Save the project.

Looking at the data editor will show that the variables with the "Save" attribute were updated to the new INIT values.

Using Unity Pro Version V5.0 (and lower)

In these versions the command "Update Local Init Values with PLC Init Values " was not available.

a. Replace the init values with the current values in the PLC (Run time) by using one of the two methods describe above.
b. Upload the project using the command "Transfer project from PLC" in the PLC menu.
c. Save the project.

Looking at the data editor will show that the variables with the "Save" attribute were updated to the new INIT values.

If using Memory Cards: This only functions using memory cards without memory Flash: this means that if "TSXMCP or TSXMFP " memory cards are being used, Initial values will not be replaced by current values. There is no warning message informing the user that the action is not done.

Appendix 2: Extract from Schneider Electric FAQ (FA315641)

What are some Best Practices for managing Unity MODICON PAC Active Data

Goals and Symptoms

The purpose of this document is to share the available options for Managing Active Data in a MODICON Unity PAC.

These include how to prevent the Loss of Data and avoid Process Downtime during PAC Application Maintenance.

Causes and Fixes

Examples of the types of Data that are most often '**At Risk Data**' include:

- Regularly Modified Setpoints (PID, Timers, Thresholds, etc.)
- Batching/Recipe Data
- Cumulative Data (e.g.: Motor Runtime, Flow Totalization, etc.)
- Tracking of Product/Process (e.g.: FIFOs, Queues, Thresholds, Transitions, etc.)

Locating this '**At Risk Data**' (@ %M/%MW) will enable the most effective options for **Data Recovery** and is the recommended method.

There are many situations that can lead to the Loss of Data, some are:

- PAC CPU failure and replacement
- PAC FW Upgrade
- PAC Application modification requiring Off-Line Build
- PAC Application Fault (WDT or anything causing HALT and required INIT)
- Accidental Off-Line Build (Changes or Rebuild All)
- Accidental Cold Start (P.S. Reset or via Unity Pro)

There are **Project Settings** that should be used to limit the user's ability to perform 'Accidental' mistakes:

- General>Project autosaving on download> save STU
  - 'download' refers to both On-Line Build Changes and Full Downloads
- General>Build Settings>Virtual Connected Mode
  - This setting disables 'Build Changes' when not connected to the PAC

Recovery from **PAC Hardware Failures and Applications Faults** will be the most challenging. These can happen at any time and if recovery of **Active Data is Critical** then it is necessary to engineer **Disaster Recovery** into the system.

MDT AutoSave with Scheduled Compare and Backups can reduce the impact of Data Loss to the frequency of the Schedule. The impact on network communications should be evaluated.

Unity Loader can provide a solution. Scripting of Unity Loader commands can be used to periodically create a '.DAT' data file of the PAC Located and Unlocated Data. However, these '.DAT' files can be useless for Unlocated Data if any type of Build is used before loading of the replacement PAC.

This leaves protecting the Data during the process of **PAC Application modifications requiring Off-Line Builds** to be addressed. There is only one method to ensure that the Data in the PAC will be the exact Data that was present in the PAC prior to the required Full Download. This procedure is:

1) Having all '**At Risk Data**' located at %M and %MW addresses
2) Having a '.STU' or '.STA' that allows connecting 'Equal' to the PAC (or Upload is possible)
3) Know that the '.BAK' file created by Unity Pro is one step back from the last On-Line Build Changes or Full Download
4) Connecting to the PAC (with above) immediately before Stopping the PAC to download the modified application
5) Performing a '**Save Data from PLC to File**' (using '.DAT/.DTX' File) and Disconnecting
6) Connecting with the modified Unity Pro Project and performing the Full Download (no RUN)
7) Performing a '**Restore Data from File to PLC**' (loading the above '.DAT/.DTX' File)
    a. Note: DTX file is recommended (requires Unity Pro V6.0 or higher)
8) Put the PAC in RUN

This procedure may not restore any or all Unlocated Data to its previous value.

There is also another method and procedure that can help to **avoid the Loss of Critical Active Data** in a MODICON PAC. This is an inherent feature of the Unity OS.

- Using the Save Parameter for Data (Variable Data and Public Variable Data of DFBs)
    a. Both of these **require frequent connection to the PAC** with Unity Pro to 'Update Local Init Values from PLC Init Values'
        i. 'Update Init Values from Current Values'
        ii. Periodic triggering of %S94
    b. Enables Off-Line comparison of '.STU' file Data with a running PAC Data via Unity DIF