Contacting Rockwell Software

Technical Support Telephone — 440-646-7800
Technical Support Fax — 440-646-7801
World Wide Web—www.software.rockwell.com

Copyright Notice

Trademark Notices

Warranty

This Rockwell Software product is warranted in accord with the product license. The product's performance will be affected by system configuration, the application being performed, operator control and other related factors.
The product's implementation may vary among users.
This manual is as up-to-date as possible at the time of printing; however, the accompanying software may have changed since that time. Rockwell Software reserves the right to change any information contained in this manual or the software at any time without prior notice.
The instructions in this manual do not claim to cover all the details or variations in the equipment, procedure, or process described, nor to provide directions for meeting every possible contingency during installation, operation, or maintenance.

Table of Contents

Chapter

1 Introduction

Welcome to RSTune, the application that makes tuning your control loops fast, easy, and accurate. RSTune also provides methods of analyzing your loops to help ensure optimal tuning parameters.

This chapter covers:

- What is RSTune™?
- RSTune and RSTune Professional Features
- System and software requirements

What is RSTune™?

RSTune is Rockwell Software's Windows®-based software for analyzing and tuning PID control loops in Allen-Bradley® PLC-5®, SLC 500™, and ControlLogix Programmable Logic Controllers, as well as the ProcessLogix hybrid control system.

Two tiers of RSTune are available: RSTune and RSTune Professional. RSTune offers complete control loop tuning. RSTune Professional adds data analysis tools and additional display options.

RSTune and RSTune Professional Features

Features of RSTune

These are some of the new features in RSTune.

- **OPC support:** RSTune is now an OPC client (RSLinx 2.1 and above only).
- **ControlLogix 5550 support added.**
- **Extra trend:** An extra trend can be added to allow you to watch another variable in the same trend.
- **Viewing of real-time trend values:** Real-time trend values can be viewed as ToolTips by positioning the cursor on the trend line.
- **View part of a Control Loop simulation:** Easily double (expand) or halve (shrink) the range on the simulation plot. Lets you view the part of the simulation that interests you most.

Chapter

3 Quick Start

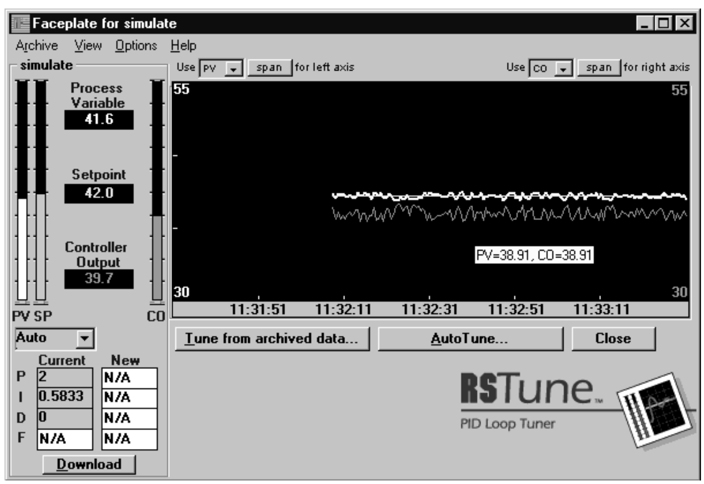
This chapter gives you a step-by-step approach to get you started using RSTune. More detailed explanations about the tuning process and how you can edit, verify, and analyze your data can be found in the remaining chapters of this User's Guide.

These topics are covered:

- Tuning a loop
- Guidelines for optimizing loops

Tuning a loop

1. Click **Faceplate** to communicate with your PID loop or software simulation. The Faceplate window is displayed.



2. Click **AutoTune**.
3. Follow the instructions on the screen to tune the loop.

Tip

For each question in AutoTune, help is available by clicking **Help**. Detailed information on AutoTune is provided in "Using AutoTune to collect data" on page 53.

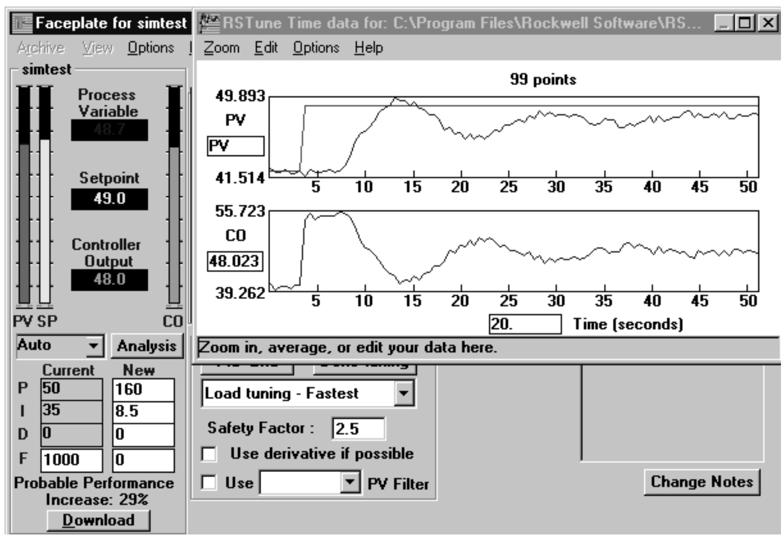
---

4. When you have completed the AutoTune sequence, RSTune displays suggested PID tuning parameters, the Time Data Window for the loop, and the safety factor, derivative, and filter information. Click **Download** to send these parameters to the processor or simulation.

   A sample of the screen after AutoTune has completed is shown here.

Tip

Filter information is not downloaded to the processor.



Chapter

5 Using RSTune

This chapter provides you with information on the use of the basic windows of RSTune, including the menu commands, displays, display options, and button functions.

These topics are covered:

- Faceplate and Trend window
- Changing the display of the Faceplate and Trend window
- Using the Off Line Analysis & PID Tuning screen
- Changing controller settings
- Debugging communications
- Menus
- Creating a report for a control loop
- Setting up extra trends
- Setting up extra loops
- Using the Extra Trends and Extra Loops

Step-by-step procedures are also provided in the online help system of the software.

The details of control loop tuning using the methods available in RSTune are covered in Chapter 6, “Tuning control loops.”

Tip

RSTune comes with a control loop simulation program that can be used to help you learn how RSTune works without being connected to a process. To use the control loop simulation:

1. In the **Choose a loop** box in the Main window, click Simulate.tun.
2. Click **Faceplate**.
3. See “Faceplate and Trend window” on page 28.

Faceplate and Trend window

The Faceplate and Trend window is the screen where you begin the process of tuning and testing your control loops.

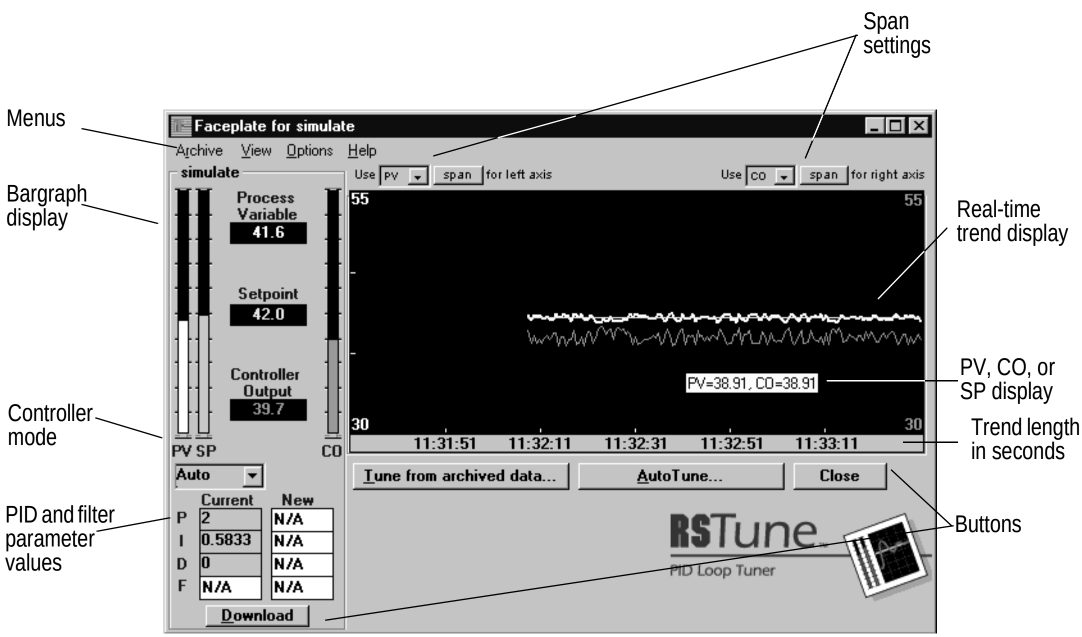
The Faceplate and Trend window displays the process variable (PV), setpoint (SP), and controller output (CO) loop variables in a bargraph, as actual values, and in trend lines. Each is the same color in each display.

Tip Controller output (CO) is sometimes referred to as the controlled variable (CV).

To display the Faceplate and Trend window:

- **From the main window:** Double-click a loop in the **Choose a loop** list.
- **From the Setup window:** Click **Faceplate**.

The screen shown here is displayed.



The Faceplate and Trend window includes:

- **Menus:** Access options and features
- **Span settings:** Allows changes to the way data is displayed
- **Real-time trend display:** Displays real-time data from your processor
- **PV, CO, or SP display:** If you hold the cursor over any point in the real-time trend display, the PV, CO, or SP values at that time are displayed. The values that are displayed depend on the Span settings. See "Changing the value of the left and right axes" on page 31.
- **Buttons:** Perform various commands
- **PID and filter parameter values:** The current processor PID values and the new values that will be downloaded to the process if **Download** is selected. InRSTune Professional, the PV filter value is also displayed. This value is not downloaded to the processor.
- **Controller mode:** The current controller mode, auto or manual.
- **Bargraph display:** Displays the loop variables in individual bargraphs and boxes.

Changing the display of the Faceplate and Trend window

You can change the display of the Faceplate and Trend window to meet your needs.

Changing the Trend display

On the Trend display, you can change:

- How ticks are displayed
- Whether the process variable, controller output, or setpoint span is used for the right and left axes
- Length of the trend (displayed as the horizontal axis of the real-time trend display)

To change how ticks and trend length are displayed:

1. Select **Options > Trend Options** in the Faceplate and Trend window.

   The Trend Options dialog box is displayed.

Chapter

6 Tuning control loops

RSTune makes analyzing, optimizing, and tuning control loops fast, accurate, and easy. You can simply follow an AutoTune sequence, or manually gather data and then have RSTune calculate the tuning parameters. You can edit the data to optimize the new parameters, and you can test the parameters without downloading them to your controller.

You can tune data online or offline. Online tuning is done from the Faceplate and Trend window. Offline tuning is done from the Offline Analysis and PID Tuning window, which is selected by clicking Offline in the main window.

These topics are covered in this chapter:

- Collecting data
- Using AutoTune to collect data
- Manually collecting data
- Using archived data files
- Using the PID and PV filter display

Collecting data

With RSTune, you can follow the AutoTune sequence to determine PID tuning parameters, or you can manually gather data and tune using that information.

Using AutoTune to collect data

AutoTune prompts you to gather data. RSTune uses the data to calculate new PID tuning parameters for your control loop.

AutoTune can be done with the controller in either Manual or Auto mode. If the controller is in:

- Manual mode: The controller output is changed
- Auto mode: The setpoint is changed

When gathering data:

- Collect the process variable and controller output data from a step or pulse test. You can make a setpoint change (in Auto) or a controller output change (in Manual).
- Both the process variable and controller output must start and end at a steady state condition and include the complete response to the setpoint or controller output change. When steady state out, both the process variable and controller output are relatively flat horizontal lines in the Trend display, moving within the range of normal process noise.
- RSTune analyzes process variable and controller output data pairs.
- All process variable filtering must be removed from the signal.

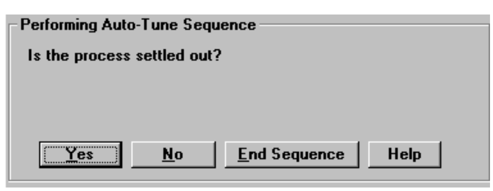
Caution The data must not be from a load or process upset. Loads must not change during the test and the range of test data should be as linear as possible. If a load change occurs during the test, click **End Sequence** and begin the test again.

When you use the AutoTune sequence, data is automatically archived.

Tip You can stop the AutoTune sequence at any time by clicking **End Sequence**.

To AutoTune:

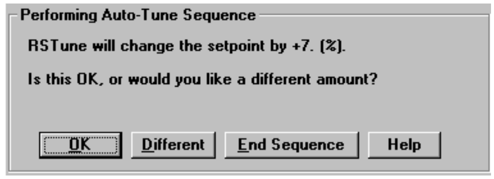
1. From the main window, click the loop to tune.
2. Click **Faceplate**.
3. Click **AutoTune**. The lower left part of the window changes as shown here.



If your loop is erratic or cycling, try:

- Putting the loop in Manual and waiting for it to settle out.
- Putting it in Auto mode and entering a low proportional gain and a low integral gain. Wait for the loop to settle out.

4. When your process data is steady state, click **Yes**. The screen shown here is displayed.



5. RSTune needs to produce a bump in your process by making a setpoint or controller output change. The default is 7.

   To use a different value, click **Different**. You are prompted for a value. You can use negative numbers if needed. Click **Enter**.

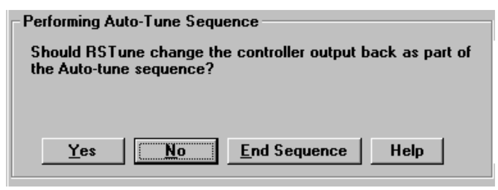
Tip

If you are using the Simulate.tun file that comes with RSTune, you might want to use a larger value. The default is not much larger than the process noise, so you will get better data if you use 10 or 15.

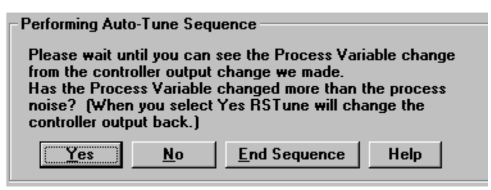
---

6. Click **OK**.
7. If the loop is in Auto mode or you are tuning a simulated loop, go to the next step.

   a. If the loop is in Manual mode, this prompt is displayed.



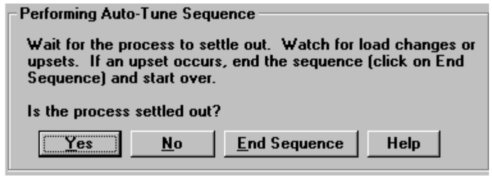
   b. If you click **Yes**, this screen is displayed.



   To get good data for tuning, RSTune needs to see the process variable respond to the controller output. The amount of process variable response needs to be at least 4 to 6 times larger than the normal peak-to-peak noise in your process.

c. After the process variable moves by this amount, click **Yes**. RSTune changes the controller output back to its original value.

This dialog box is displayed.



8. When the process has steady state, click **Yes**.
9. The name of the archive file for this data is displayed. Click **OK**.
10. The Time data window for the data is displayed (see Chapter 7, "Using the Time data window"). You can start verifying or editing your data and determining new PID tuning values.

Manually collecting data

Requirements for gathering valid data:

- Collect the process variable and controller output data from a step or pulse test. You can make a setpoint or controller output change.
- Both the process variable and controller output must start and end at a steady state condition and include the complete response to the setpoint or controller output change.

Caution

Plant data taken for RSTune analysis and tuning must have all process variable filters removed from the signal.

Important

The data must not be from a load or process upset. Loads must not change during the test and the range of test data should be as linear as possible. If a load change occurs during the test, stop collecting data and start over.

Collecting data manually

In some cases, you might want to collect data manually instead of using the AutoTune sequence. This is a basic procedure to collect data manually. This is a closed loop test with the controller in the Auto mode.

Chapter

8 Using the output Characterizer

Available only with RSTune Professional.

RSTune provides both output and pH (input) Characterizers. This chapter only discusses the output Characterizer. See Chapter 9, "Using the pH Linearizer," for information on the input Characterizer.

Many control loops are difficult to tune because they are non-linear. This means that the process gain changes as a function of the measurement or controller output. Without any linearization, the controller needs to be tuned for the condition where the process gain is the highest. This results in sluggish tuning everywhere else. Linearizing these loops improves control because the controller is better tuned over the complete operating range.

The output Characterizer takes the output of the controller and transforms it to a value so that the control loop is linear over the entire range of control. A control loop with a characterizer should allow you to have robust optimal tuning parameters anywhere over the CO or PV range.

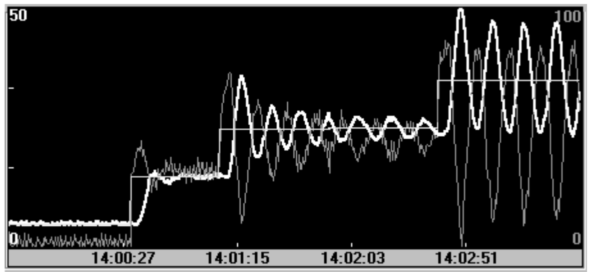
Use the output characterizer:

- if servo control is important
- if you need to linearize flow loops, jacket temperature in split-range chemical reactors, or slave loops in cascades
- for any loop where the setpoint will change
- if the linearity of the loop changes with load or production rate

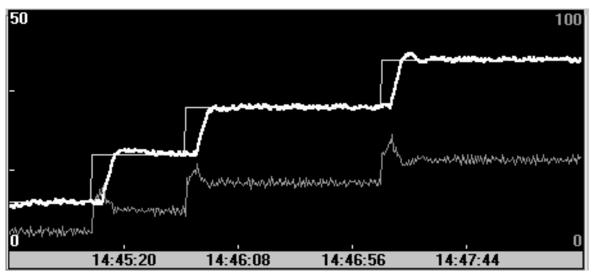
Characterizer can supply code in FORTRAN, Basic, C, or X-Y line endpoints. If you are using RSLogix Frameworks, you can download the X and Y values directly to the Lin function. Use the code to program the PID loop in your ladder logic. See page 97 for ladder logic examples.

Example of when to use the output Characterizer

A typical example of when to use the output Characterizer is shown here. This process oscillates toward one end of the range and has sluggish response at the other.



With a characterizer, you might be able to get uniform performance across the entire range of your process. This allows your process to run optimally without de-tuning for oscillations.



Characterizer tests the final control element with the rest of the process to guarantee a good fit between the characterization curve and the installed characteristic of the final control element and process.

Building a characterization curve

Before using the output Characterizer, make sure the controller is tuned to stabilize the loop at that value of controller output where the loop gain is highest.

You need to complete these steps to use Characterizer:

- Collect process data
- Start Characterizer
- Run the Build Characterizer Wizard

Collecting process data

You need to collect some process data at different operating points in either auto or manual. If you collect data in auto mode, you will bump the setpoint rather than the controller output.

1. On the Faceplate and Trend window, select manual or automatic control.
2. Select **Archive > Archive On**.
3. In:
   - Auto mode: Change the setpoint to 10% of the process variable.
   - Manual mode: Change the controller output to 10% of the process variable.
4. Let the process reach steady state.
5. Increase the setpoint (manual) or controller output (manual) by 15% of the initial controller output.
6. Let the process settle out (reach steady state).
7. Repeat steps 5 and 6 until you have collected data at 7 data points.
8. Select **Archive > Archive Off**.

Starting Characterizer

1. On the Faceplate and Trend window, click **Tune from archived data**.
2. Select the archive file you just created.
3. Click **Tune**.
4. In the Time data window, select **Options > Characterizer**.
5. Place a vertical line over each steady state area and double-click. This places a line at that point, as shown in the example below.

Tip

RSTune saves your characterization curve. Therefore, if you have already built a characterization curve for this archive, the lines are already placed. You can go directly to the Build Characterizer Wizard or adjust the line positions first.

Chapter

10 Control loop analysis

RSTune provides advanced analysis tools that you can use to test your tuning parameters before downloading them to the controller. Four of the analysis tools are available by clicking **Analysis** on the Faceplate and include:

- Process Model
- Process Frequency Response Plot
- Control Loop Simulation Plot
- Robustness Plot

These tools can be used to do "what if" analysis of your tuning parameters before you download them to the controller.

RSTune Professional also provides these analysis tools:

- Power Spectral Density plots
- Auto- and Cross-Correlation

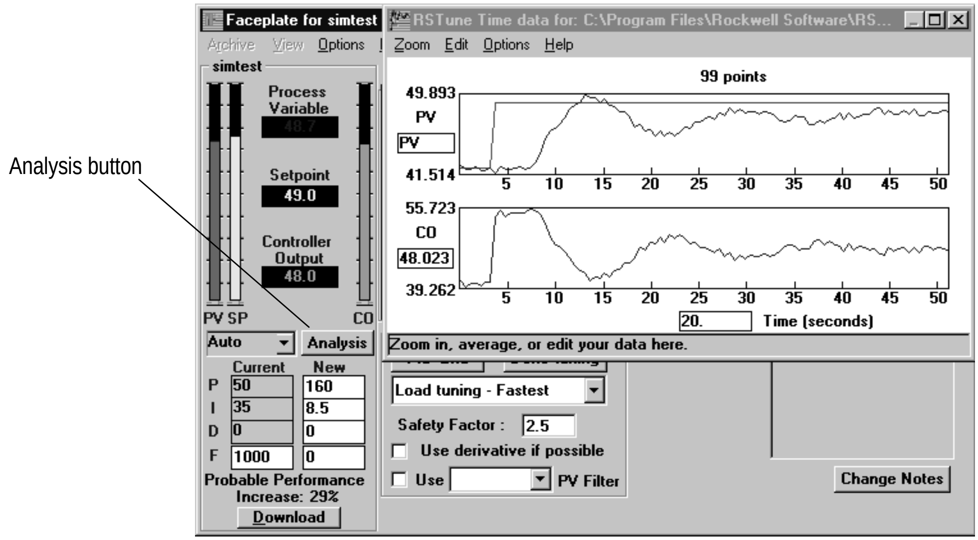
Tip

The quality of the information in the analysis windows depends on the quality and relevance of the data that you collected. RSTune models are based on frequency response. Frequency response is based on the data shown in the Time data window.

Using the standard analysis tools

To use the Process Model, Process Frequency Response, Control Loop Simulation, and Robustness plots:

1. Open the Faceplate.
2. Click **Tune from archived data**.
3. Select the archive file to tune.
4. Click **Tune**. The Time data window is displayed.



5. Click **Analysis**.

   All four analysis plots open.

Tip

If you close one of the analysis windows, all of the analysis windows close.

When you change a PID tuning value, the Control Loop Simulation and Robustness plots update immediately. (See “Robustness plot” on page 128 for information on the Robustness plot.)

---

6. Begin working with the plots by selecting a model from the Process Model window.

Selecting a process model

The Process Model window shows the process model type identified for your control loop. RSTune uses your process frequency response to identify the model as first order or second order.

Tip

The quality of your models, simulation, and robustness depend on the quality and relevance of your time data.

To select a process model:

1. Adjust the plot windows so that you can see the Process Model, Process Frequency Response, and Robustness Plot.
2. Check the data on the Process Frequency Response and Robustness Plots.
3. In the Process Model window, select the other Model type from the list (**Force steady state gain** or **Allow gain to float**). The plots change to reflect the new model.
4. Check the data on the Process Frequency Response and Robustness Plots.
5. In the Process Model window, select the Model type that fits your data the best.

Tip	The Process Frequency Response plots both the actual and the model data. Choose the model that is closest to the actual data of Amplitude Ratio and phase. The most important frequencies are where the phase is between about 90 and 180 degrees (phase lead) or –90 and –180 degrees (phase lag). The Robustness Plot shows the trade-off between tight tuning and sensitivity to process changes. If you have two sets of tuning with about the same robustness, you can generally choose the faster response.

6. Once you've determined the best Process Model, you can use the Process Simulation window and the Robustness Plot to analyze the calculated and current PID parameters.

Appendix

A DDE topics

RSTune requires a DDE topic for each control loop you need to communicate with.

What is a DDE topic?

A topic is a path from Windows' Dynamic Data Exchange (DDE) to a processor. When a DDE-compatible Windows application makes a DDE request for information from a communication package such as RSLinx, it must specify a topic name and an item. The information stored in the topic determines:

- Which communications Driver Type and Driver Number to use
- Whether to access a local or remote station
- What station number to access
- What kind of processor is at that station number
- How to communicate with the processor: poll or unsolicited
- How long to wait before registering a time-out error

You can also have multiple topics to communicate with several different processors at once. The minimum number of topics needed in a project is one for each different processor with which you wish to communicate. For example, you might have a topic named Poll 25 that communicates with a PLC-5 processor at station 25 and a topic named Poll 18 that communicates with a PLC-5/250 processor at station 18.

Multiple topics can be used in reference to the same processor. One topic called Poll 25 might poll station 25 for data. Another topic called Wait 25 might wait for an unsolicited message from station 25. You could even have additional topics that poll station 25 at different poll rates.

Tip Different topics can be used simultaneously, and different DDE-compatible Windows applications can use the same topics simultaneously.

Glossary

Activation disk. Any disk (floppy or hard) containing an activation file. An activation disk can be used to activate the software. This is different than a key disk (Master Disk) in that at least one license of the software must be available on the activation disk to activate the software.

Activation file. A hidden read-only system file that "activates" a Rockwell Software product. The software will run properly only if your system can find the correct activation file.

Activation key. Activation files contain a database of activation keys. Each key is particular to a certain product and must be accessible on a local or remote drive for that product to run.

Archiving. The action of saving data collected from a particular control loop to the hard drive. This data file is saved to the same directory as your control loop (.tun) file.

Auto mode. In Auto mode, the controller or processor automatically controls the output commands based on its program. This is a closed loop.

AutoTune. The RSTune AutoTune sequence automates the process of collecting data for PID tuning. Answer the questions to the RSTune AutoTune sequence and RSTune finds optimal tuning for your loop.

Averaging filter. The averaging filter is a moving average filter — every sample interval it gives the average of the previous time constant of values. The effect of the time constant is about half that of the first order filter. To get the equivalent first order time, use an averaging filter with an averaging window length of twice a first order constant. The averaging filter is a good filter for reducing frequencies of noise occuring at the filter time. If you have a specific frequency you want to remove, setting the average filter time equal to the period of that noise should completely remove it (period is the inverse of frequuency). In general, it is not as good as the butterworth at higher frequencies.

Butterworth filter. *See* Second order Butterworth filter.

Bump test. *See* Step test

Cascading loops. The output of one control loop manipulates the setpoint of another control loop. Includes a master loop and a slave loop(s).

Index