

Communication Settings

The Galil SmartTERM application installation (as well as WSDK, ActiveX, and DMCWIN32 installations) includes the necessary drivers and .DLL files required to communicate with the Galil controller. The drivers are automatically installed and default communications settings are applied to the device by the driver when a card is installed as per the installation procedure outlined in Ch.2. However, some advanced settings are available to modify the communications methods and data record access. These settings are accessed through the Galil Registry Editor after the card is properly installed.

Galil Registry Editor

The “Edit Registry” dialog box (shown in Fig 4.4) can be accessed by selecting **Controller Registration...** under the **Tools** menu (or by selecting the toolbar icon with the magnifying glass) within DMC SmartTERM. The Edit Registry dialog shows the current controller models installed to the PC along with their associated I/O addresses, interrupt lines, and controller serial numbers. The Galil Registry is part of the DMCReg.ocx ActiveX object (refer to Fig 4.4). This ActiveX control is used to create, maintain, and modify the communication parameters, which are discussed next.

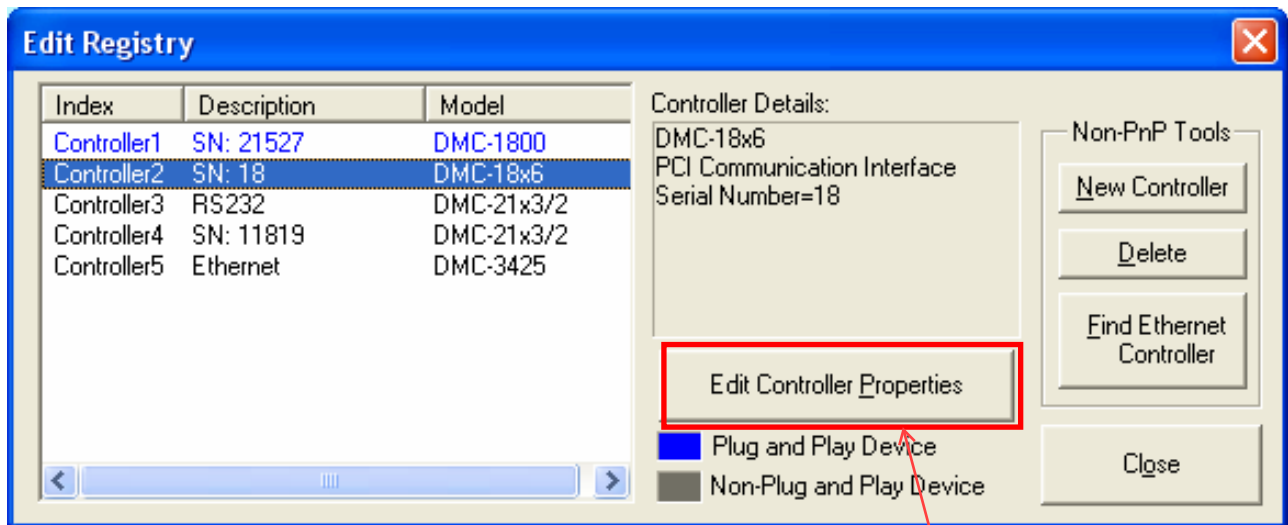


Figure 4.4 - Galil Registry Editor

Select your controller (there will be only one in most Winspect systems) and click on Edit Controller Properties

Setting Communications Parameters and Methods

To access the Controller Communication Parameters dialog, highlight the desired controller in the Galil Registry Editor accessed through SmartTERM and select the **Properties** command button.

The timeout property under the **General Parameters** tab (shown in Fig 4.5) allows the user to select the timeout period that the Galil software waits for a response from the controller before generating an error. If the controller does not reply to a command with the data response and a colon (or just a colon for commands that do not invoke responses), then the Galil software API will generate the timeout error code -1 (A time-out occurred while waiting for a response from the Galil controller). The default setting for the timeout is 5000ms, which should be sufficient for most cases.

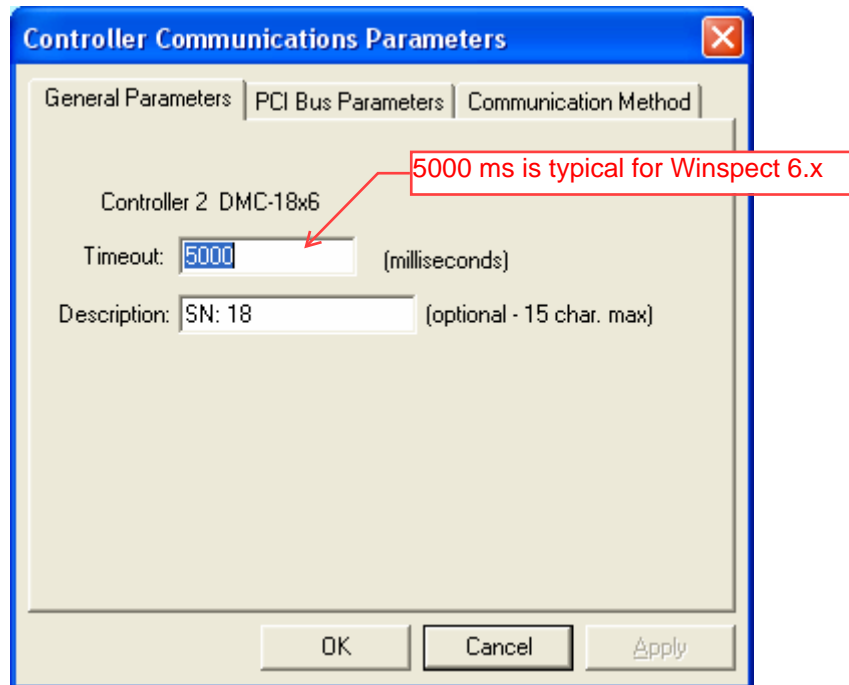


Figure 4.5 - General Communications Parameters Dialog

Advanced communications settings are available under the **Communications Method** tab to allow different methods of communications to be utilized (shown in Fig 4.6). The version 7 (and higher) drivers and .DLL's allow for three different methods of communications: Interrupt, Stall, and Delay.

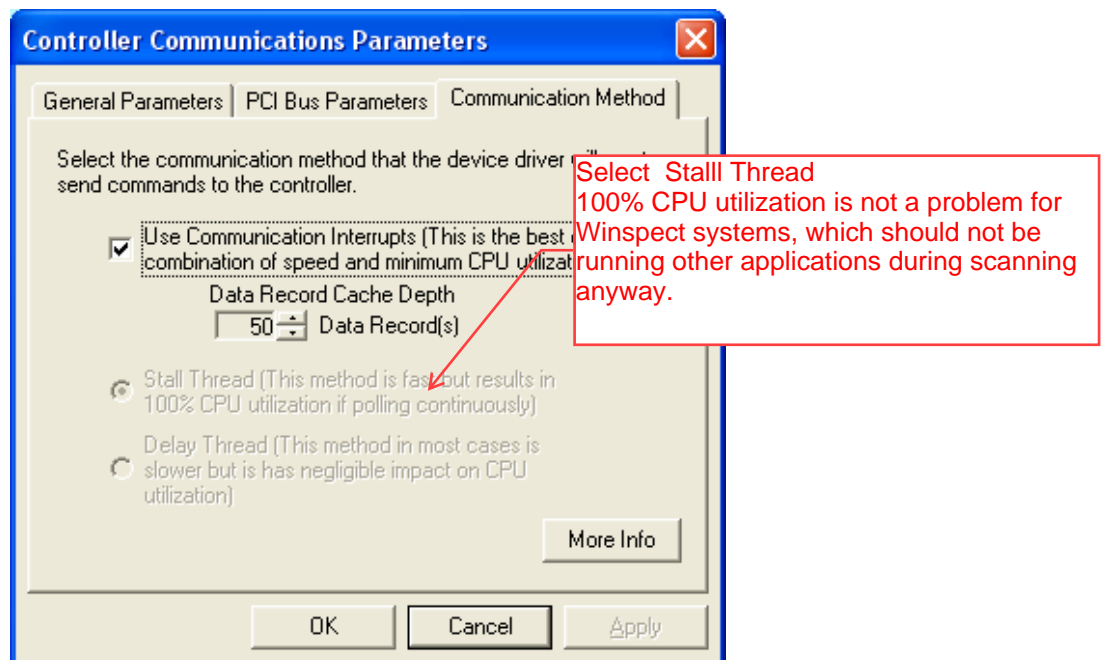


Figure 4.6 - Controller Communications Method Dialog Box

Interrupt Communications Method

The interrupt method overall is the most efficient of the three methods. The interrupt communications method uses a hardware interrupt to notify the driver that a response or unsolicited data is available. This allows for greater efficiency and response time, since the drivers do not have to “poll” the buffers for the data. Additionally, the interrupt method allows for data record caching.

The interrupt method uses bus level interrupts (IRQ) from the controller to notify the PC that data is available. This requires that the Controller be configured with a valid interrupt line. For DMC-18x6 controllers the interrupt is configured automatically. For complete information on the different communications methods, select the **More Info** button on the Communications parameters dialog box.

Data Record Cache Depth

With the “interrupt communications” method enabled, the driver will cache data records for retrieval via API function calls. This makes it possible to not 'miss' any data records, even if the DR command has been configured to refresh the data record every two milliseconds. For example, a program could poll at a relatively long frequency (say every 50 milliseconds), and not miss any data. The cache depth can be set when the interrupt communication method is selected. The data record cache functions like a FIFO. Reading the data records removes them from the cache. If the cache is full and a new data record arrives from the controller, the new data record is placed in the cache and the oldest data record in the cache is discarded. If multiple handles to a controller are open, the first handle to retrieve the data record(s) will possess the only copy available. When an application needs only the most recent data record available, the cache depth should be set to 1.

Stall Thread and Delay Thread Methods

Users can also choose between "Delay" and "Stall" methods. These two methods are available for both the DMC-18x6 controllers and affect how the software "waits" for a response from the controller when a command is sent. If a controller is configured with the "Delay" method, the thread waiting for a command response gives up its time slice, allowing other processes running on the operating system to proceed. This method can slow communication, but results in negligible CPU utilization. The second method, the "Stall" method, uses the opposite strategy. The thread that performs I/O with the controller maintains ownership of the CPU and polls the controller until a response is received. This approach is essentially the same method employed in previous versions (< V7) of the Galil communication DLLs and drivers. While the "Stall" method does not have to wait for its thread to become eligible for execution, it does result in [100% CPU utilization](#) while communicating with the controller.

Data Record Refresh Rate

Under the **PCI Bus Parameters** tab, the rate at which the data record is sent to the software drivers can be configured. The period between refreshes can be set from 2 - 256 ms (assuming the standard TM setting of 1000 is set). The Galil communications .DLL will use this value to send the appropriate DR command to the controller when a communications session is opened.