

# Using ioDisplay 6.0 with SNAP High-Density Digital Modules

## Introduction

If you are running ioDisplay 6.0 and using SNAP high-density digital (HDD) modules, you'll need to take an extra step in order to monitor or change module data in your ioDisplay project. Both monitoring and changing module data requires that your ioControl 6.1 strategy read HDD point data into variables that are then read or changed by ioDisplay.

**IMPORTANT:** *When ioDisplay 6.1 is released, this extra step will no longer be needed. If possible, we recommend that you wait for ioDisplay 6.1 rather than using the extra programming steps described in this technical note.*

This technical note describes how to set up the necessary variables in ioControl 6.1 and read and write to them from ioDisplay 6.0. This technical note assumes that you already have ioControl 6.1 and ioDisplay 6.0 installed on your computer and that you are familiar with using them. For information on these two software products, see the following guides:

For using	See these guides	Opto 22 form
ioControl	<i>ioControl User's Guide</i> <i>ioControl Command Reference</i> <i>ioControl Commands Quick Reference Card</i>	1300 1301 1314
ioDisplay	<i>ioDisplay User's Guide</i>	1302

## For Help

If you are having trouble using the sample files and cannot find the help you need in this technical note or in the guides listed above, contact Opto 22 Product Support.

**Phone:** 800-TEK-OPTO (835-6786)  
951-695-3080  
(Hours are Monday through Friday,  
7 a.m. to 5 p.m. Pacific Time)

**Fax:** 951-695-3017

**Email:** support@opto22.com

**Opto 22 Web site:** support.opto22.com

*NOTE: Email messages and phone calls to Opto 22 Product Support are grouped together and answered in the order received.*

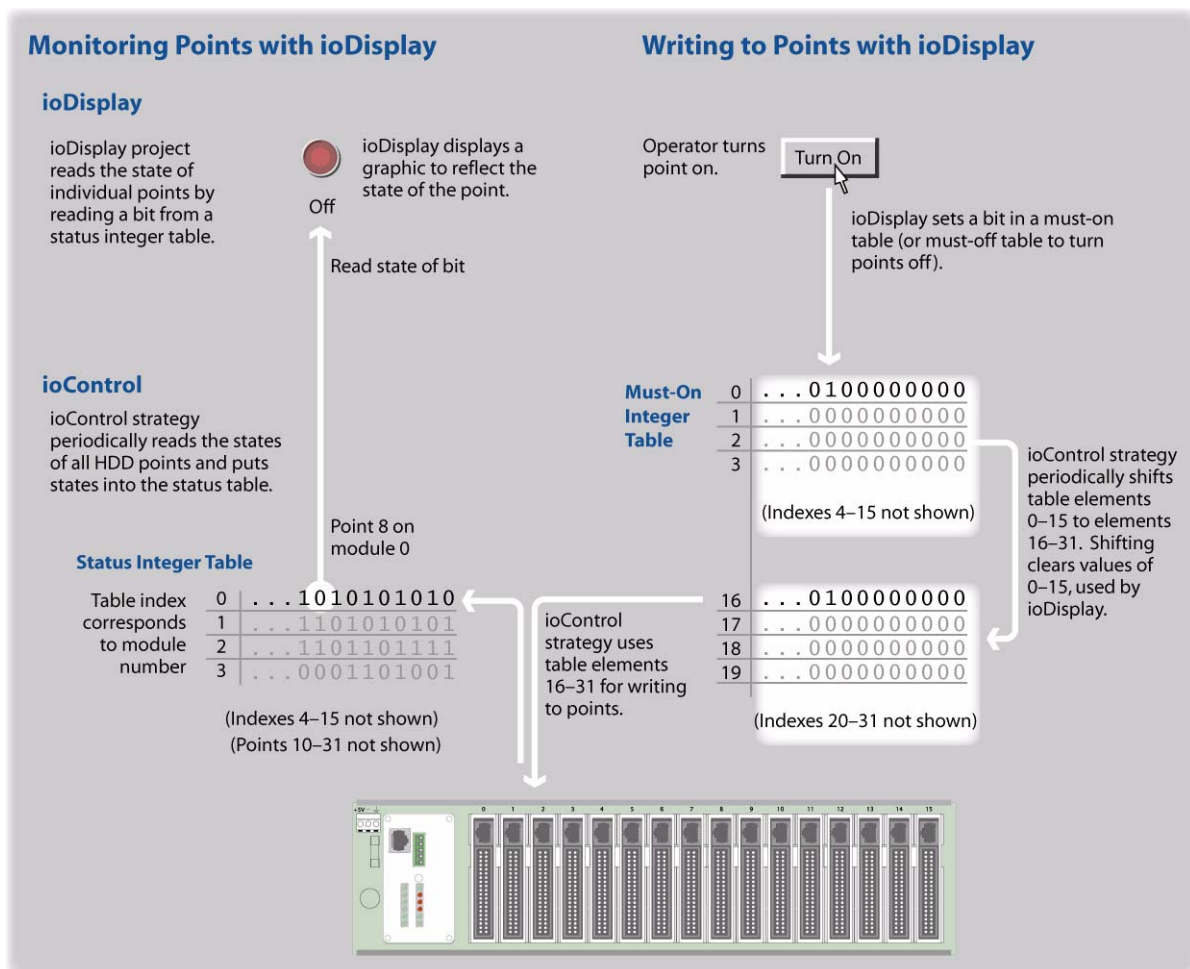
## Example: Using HDD Modules with ioDisplay 6.0

The graphic below shows an example of monitoring and writing to HDD modules from ioDisplay, using three table variables as part of the ioControl strategy.

**Monitoring**—ioControl periodically reads the on/off status of every point on every HDD module on the rack and places the data into an integer 32 table. ioDisplay uses the data in this status table to update its display.

**Writing**—When an operator changes an HDD point, for example by pushing a button on the display, the instructions are sent to one of two integer 32 tables: one for turning points on (must-on) and one for turning points off (must-off). Each of these tables is twice as big as it needs to be to hold the data, because the data is placed in the first half and periodically shifted to the second half. The shift command temporarily locks the table so that write commands are not missed or confused. ioControl then uses the data in the second halves of both tables as MOMO masks to write to points.

The complete process is illustrated below.



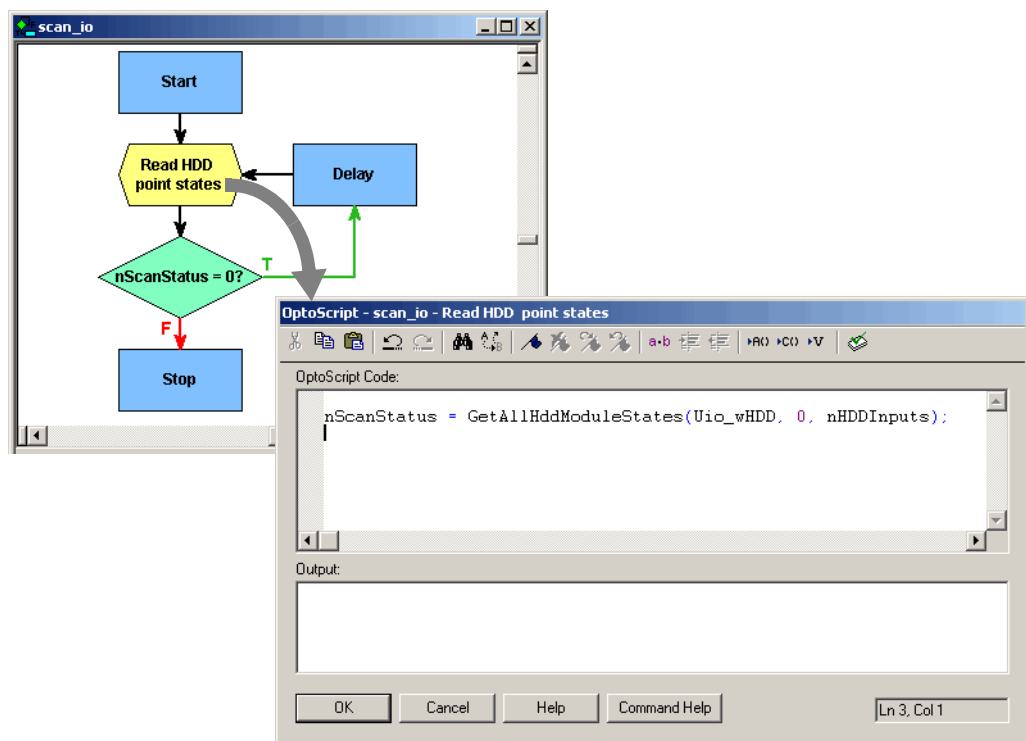
## Table Variables in the ioControl Strategy

As an example, suppose that in your ioControl strategy, you have one chart (called Scan\_IO) to handle reading the HDD points and another chart (Write\_IO) to handle writing from ioDisplay.

### Scan\_IO Chart

The Scan\_IO chart uses the command Get All HDD Module States to read the on/off states of all HDD points on the rack and place the data in a table variable, called nHDDInputs. This table must contain at least 16 elements including the starting index, because the Get All commands return 16 elements (corresponding to the largest rack size) regardless of the actual size of the rack.

As shown below, any error is placed in the variable nScanStatus.

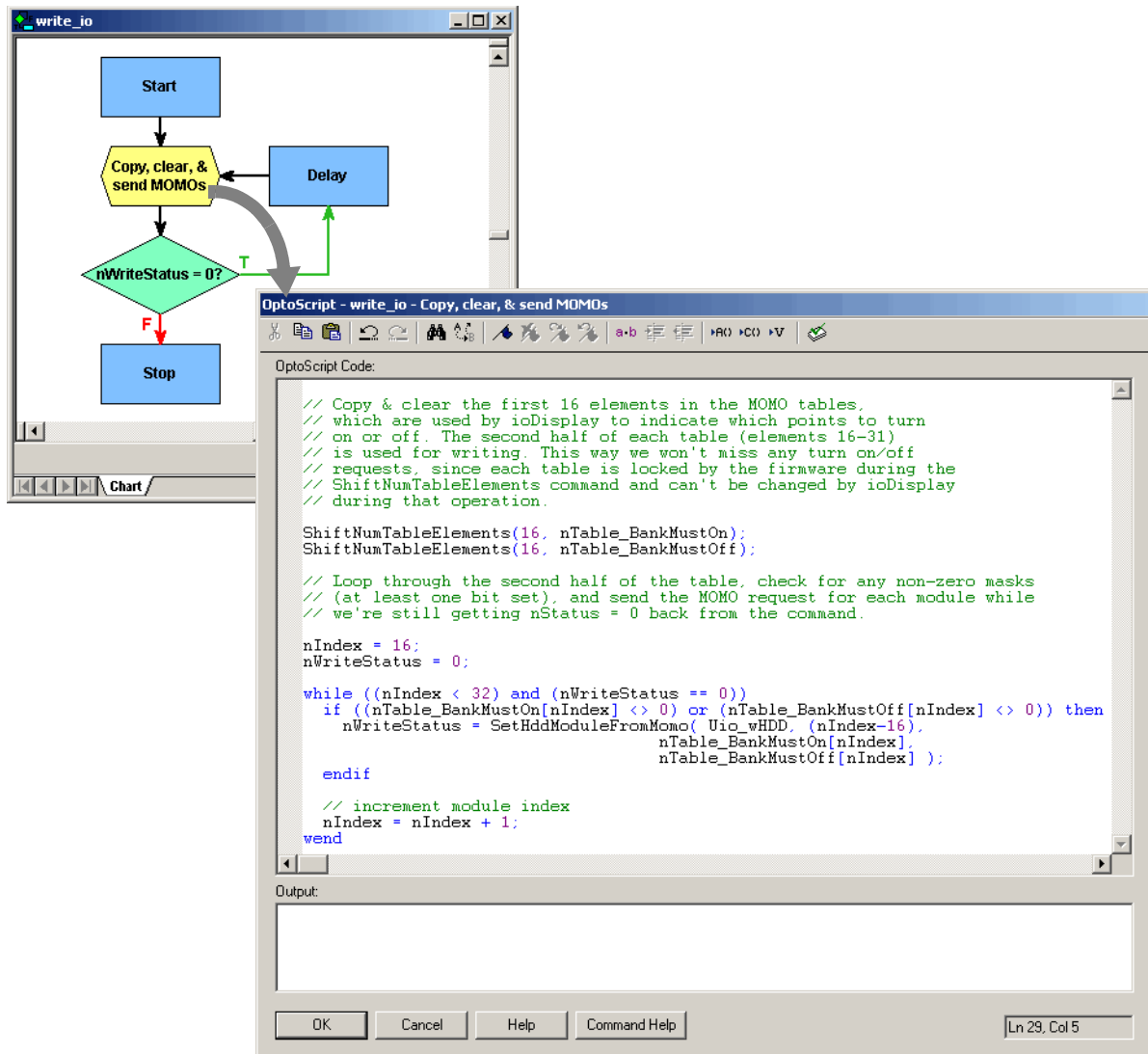


The chart checks for an error in nScanStatus and stops the process if one is found. Errors should be handled and the process restarted as appropriate for your application.

## Write\_IO Chart

The Write\_IO chart handles the task of writing to points based on an operator's interaction with ioDisplay. In this chart you see two table variables, nTable\_BankMustOn and nTable\_BankMustOff, used as must-on and must-off tables for the entire rack.

These tables each contain 32 elements. Elements 0–15 hold the write commands coming from ioDisplay. The data in this first half of the table is periodically moved to elements 16–31, which then become the on-mask or off-mask that points are set from.



## In ioDisplay

In ioDisplay Configurator, when assigning dynamic attributes to elements in the display, reference the ioControl variables you created to read or write to HDD points.