

FactoryFloor to ioProject Migration

Introduction

This technical note describes how to migrate a control system that uses Opto 22 FactoryFloor[®] software applications—specifically OptoControl[™] and OptoDisplay[™]—to Opto 22's ioProject[™] Professional software. The following sections are included:

- [“Preparing for Migration”](#) on page 5
- [“Converting OptoControl Strategies to ioControl”](#) on page 6
- [“Converting OptoDisplay Projects to ioDisplay”](#) on page 23

NOTE: Customers moving from legacy Cyrano software need to migrate from Cyrano to FactoryFloor before moving to ioProject. Follow steps in Opto 22 form #1169, Tips for Cyrano Users, available on our website, www.opto22.com.

Why Migrate to ioProject?

ioProject Professional software and Opto 22 SNAP PAC[™] controllers provide a migration path for legacy control systems based on FactoryFloor software and *mistic* controllers. Moving to contemporary hardware and software can expand your control system's capabilities while maintaining your investment in I/O units, sensors/actuators, wiring, and other components.

ioControl[™] Professional and ioDisplay[™] Professional software applications are designed for more complex projects, especially those requiring OPC (OLE for Process Control), multiple protocol, multiple network, or legacy hardware support. ioProject Professional is designed for use with SNAP PAC controllers and takes advantage of their features, including dual independent Ethernet network interfaces for redundant Ethernet links or segmented networking. See the section [“Comparing Opto 22 FactoryFloor and ioProject Professional”](#) on page 2 for specific features and more information.

Comparing Opto 22 FactoryFloor and ioProject Professional

The following table compares the features of ioProject Professional and FactoryFloor software suites. Also see Opto 22 form #1485, the *SNAP Controller Comparison Chart*, for details on the capabilities of different Opto 22 controllers.

Feature	ioProject Professional	FactoryFloor
Included software	<ul style="list-style-type: none"> • ioControl Professional • ioDisplay Professional • OptoOPCServer™ • ioManager™ • utilities 	<ul style="list-style-type: none"> • OptoControl • OptoDisplay • OptoServer™ • OptoUtilities™
Network	<ul style="list-style-type: none"> • <i>To host</i>: Ethernet/TCP or PPP over serial • <i>To I/O</i>: S-series—Ethernet/UDP and serial; R-series—Ethernet/UDP only • Serial or Ethernet/TCP to third-party devices • Support for Ethernet link redundancy or segmented control network 	<ul style="list-style-type: none"> • <i>To host and I/O</i>: Serial, Ethernet, or ARCNET (Ethernet only on M4 controllers with expansion card. ARCNET on M4 or G4LC32SX only with card; ARCNET is built into G4LC32) • Serial to third-party devices
I/O unit compatibility	<ul style="list-style-type: none"> • SNAP PAC R-series I/O units (R-series also have built-in controllers) • SNAP Ultimate I/O™ units • SNAP Ethernet I/O™ units • SNAP Simple I/O™ units • E1 and E2 I/O units • Serial <i>mistic</i> I/O units (SNAP PAC S-series controllers only): B3000, SNAP-BRS, B100, B200, G4D16R, G4D32RS, G4A8R 	<ul style="list-style-type: none"> • <i>mistic</i> protocol I/O units including <i>mistic</i> bricks, B3000, B100, B200, and high-density analog • SNAP Ethernet and SNAP Simple I/O units (M4 bus controllers only; requires Ethernet card) • ARCNET I/O units (G4LC32; or M4 bus or G4LC32SX controllers with ARCNET card)
Control software		
Name	ioControl Professional	OptoControl
Controllers used	<ul style="list-style-type: none"> • SNAP PAC S-series • SNAP PAC R-series 	<ul style="list-style-type: none"> • SNAP-LCM4 • SNAP-LCSX • SNAP-LCSX-PLUS • G4LC32 series • M4 and M4RTU series
Main features	<ul style="list-style-type: none"> • Flowchart programming • OptoScript™ programming • Subroutines (debuggable) • Graphical debugger • Additional features on SNAP Ethernet brains (ramping, etc.) • Conversion utility for OptoControl strategies (version 4.1) • Support for serial <i>mistic</i> I/O units • Ethernet link redundancy 	<ul style="list-style-type: none"> • Flowchart programming • OptoScript programming • Subroutines (no debugging within strategy) • Graphical debugger • Cyrano™ conversion utility • Support for serial <i>mistic</i> I/O units
Maximum charts running at once	<ul style="list-style-type: none"> • 32 on SNAP PAC S-series (plus host task) • 16 on SNAP PAC R-series (plus host task) 	32 (including host task)

Feature	ioProject Professional	FactoryFloor
Proportional-integral derivative (PID) loops	<ul style="list-style-type: none"> • 4 PID algorithms for Ethernet (32 PIDs per SNAP Ultimate I/O unit; 16 PIDs per SNAP Ethernet I/O unit) • 1 PID algorithm for serial (8 PIDs per <i>mistic</i> I/O unit) • Graphical tool for manual tuning of Ethernet and <i>mistic</i> PIDs 	<ul style="list-style-type: none"> • 1 PID algorithm • 8 PIDs per brain
Ethernet link redundancy	<ul style="list-style-type: none"> • Primary and secondary IP addresses for SNAP PAC controllers and I/O units • ioControl commands can be used to control redundancy algorithm 	--
Additional toolkits	<ul style="list-style-type: none"> • Allen-Bradley DF-1 Integration Kit • OptoMMP™ Communication Toolkit 	<ul style="list-style-type: none"> • Allen-Bradley DF-1 Integration Kit • Modbus Integration Kit • OptoControl SDK
HMI software		
Name	ioDisplay Professional	OptoDisplay
Main features	<ul style="list-style-type: none"> • Alarming • Trending • Security • 3000-graphic library • Conversion utility for OptoDisplay projects • Ethernet link redundancy • Multiple OPC server support (OptoOPCServer) 	<ul style="list-style-type: none"> • Alarming • Trending • Security • 3000-graphic library • Conversion utility from <i>mistic</i> MMI
Controllers supported	<ul style="list-style-type: none"> • ioProject controllers • FactoryFloor controllers on Ethernet network 	<ul style="list-style-type: none"> • FactoryFloor controllers • Cyrano controllers
Ethernet link redundancy	<ul style="list-style-type: none"> • Primary and secondary controller • Primary and secondary scanner 	--
OPC server		
Name	OptoOPCServer	OptoServer
OPC version	OPC 2.0-compliant	OPC 1.0-compliant
Ethernet link redundancy	ioDisplay primary and secondary controllers	--

Opto 22 FactoryFloor and ioProject Software Applications

The following table matches an Opto 22 FactoryFloor software application to its comparable application available in ioProject.

FactoryFloor	ioProject
OptoControl	ioControl
OptoDisplay	ioDisplay
OptoTerm	ioManager™, ioTerminal™
OptoUtilities™ OptoSnif™ OptoTerm™ OptoVersion™	ioUtilities™™ ioMessageViewer™ ioTerminal™ OptoVersion™ OptoENETSniff™ 1
OptoServer	OptoOPCServer
OptoConnect	ioSQL™ 2

1 OptoENETSniff is a software tool used to monitor, or “sniff,” traffic on an Ethernet network. OptoENETSniff includes features for easily identifying Opto 22 devices operating on an Ethernet network.

2 ioSQL is planned for future release. Contact Opto 22 Product Support for current information.

Documentation

See the following guides for additional information on using ioControl, ioDisplay, and related software applications:

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
OptoOPCServer	OptoOPCServer User's Guide	1439
ioManager	<i>ioManager User's Guide</i>	1440

For Help

If you are having trouble migrating your control system to ioProject Professional and cannot find the help you need in this technical note or in the related documentation, 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 website: www.opto22.com

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

Preparing for Migration

Requirements

- **ioProject Professional software suite**—The FactoryFloor controllers used in the control system must be configured on the PC used for converting to ioControl and ioDisplay.
- **Opto 22 FactoryFloor R4.1 software suite**—If you have a version of FactoryFloor that is older than R4.1, contact Opto 22 Product Support for assistance.
- **SNAP PAC controller(s)**—The controller(s) should already be configured and connected to an Ethernet network. See the user's guide for the controller if necessary.

1—Inventory Your Existing Control System

Before starting to convert OptoControl strategies and OptoDisplay projects, we recommend that you list or map the hardware and software components of your control system. Doing this will help to identify unsupported hardware, functions that are not supported in ioProject, and other potential obstacles to migrating your system.

Make sure to identify all controllers, I/O units, and the type of network (Ethernet, RS-485, ARCNET, etc.) used to connect them. Also list any PCs running OptoDisplay or other FactoryFloor applications.

2—Back Up Your Current Strategies and Projects

Before making any changes to your existing OptoControl strategy or OptoDisplay project, back up all files to one or more secure locations.

3—Convert Strategies and Projects to FactoryFloor R4.1

ioControl Professional and ioDisplay Professional can only import strategies and projects saved in OptoControl R4.1 and OptoDisplay R4.1, respectively. If necessary, open older strategies and projects using OptoControl R4.1 and OptoDisplay R4.1 and then save the files. If you have an older version of FactoryFloor, contact Opto 22 Product Support for assistance.

For Cyrano and *mistic* MMI Users

Legacy Cyrano control programs or *mistic* MMI projects must be saved in FactoryFloor R4.1 before starting the conversion steps described in this technical note. There are functional differences between Cyrano and OptoControl strategies that you must consider when opening a Cyrano strategy in OptoControl. These differences and other information are described in *Tips for Cyrano Users* (Opto 22 form 1169), which can be downloaded from the Opto 22 website (www.opto22.com). Also see the following documents for more information on using Cyrano and *mistic* MMI in OptoControl and OptoDisplay:

Application	See these guides	Opto 22 form
Cyrano	<i>OptoControl User's Guide</i> <i>OptoControl Command Reference</i>	724 725
<i>mistic</i> MMI	<i>OptoDisplay User's Guide</i>	723

Converting OptoControl Strategies to ioControl

ioControl Professional handles the OptoControl-to-ioControl conversion automatically. However, depending on the complexity and features of the OptoControl strategy, you may need to change or adjust the strategy both before and after conversion.

1—Assessing Major Differences in Features

Before converting an OptoControl strategy into ioControl, determine if any features and functions used in the OptoControl strategy are unavailable or implemented differently in ioControl. For most features and functions, there are alternative approaches or workarounds to accomplish the same task using ioControl. On [page 7](#), review the table of OptoControl features not available in ioControl; also check "[Additional OptoControl and ioControl Differences](#)" on [page 7](#).

OptoControl Features Not Available in ioControl

OptoControl Feature Not Available in ioControl	Alternative, Workaround, or Notes
Multiple RS-485 serial ports	Only one serial port is available on SNAP PAC controllers for communication with <i>mistic</i> remote bricks and other serial-based <i>mistic</i> I/O units. If the OptoControl strategy references serial I/O units on more than one serial port, combine the ports into one and place a non-terminated SNAP PAC controller in the middle of the serial link. Make sure that other devices on the serial network do not use the same address.
Serial host support (except for PPP over a modem)	Use PPP and a modem.
ARCNET networking	Evaluate your networking environment to see if Ethernet or serial networking is an option.
<i>mistic</i> signal interrupts	<i>mistic</i> signal interrupts are not supported. Check the Opto 22 website (www.opto22.com) or contact Opto 22 Product Support for current information on signal interrupt support.
External instruction sets (also called “XIDs” and “Forth libraries”)	Most functions implemented using external instruction sets have been integrated into ioControl. See “ OptoControl and ioControl Commands ” on page 16 for information on whether a specific function is supported. If necessary, recreate what the external instruction set does OptoScript and/or subroutines.
ASCII/checksum support for <i>mistic</i> brains	ioControl only supports binary/CRC format for <i>mistic</i> brains. Contact Opto 22 Product Support for current information on ASCII/checksum options in ioControl.
HRD analog brick support	Consider using a SNAP I/O unit with 4-channel SNAP analog modules. Up to 16 SNAP analog modules can be placed on one SNAP rack, for up to 64 analog points.
Localbus support	Change the localbus port in OptoControl to a remote (serial) port. Resolve any potential I/O address conflicts if you are already using serial ports. You will need to replace the local I/O unit and points with compatible replacements. Also evaluate your application and networking environment to see if using Ethernet-based brains is an option.
RS-232 port used with RS-485 converter for <i>mistic</i> I/O	<i>mistic</i> serial I/O units must use the single RS-485 port on a SNAP-PAC-S1 controller. Multiple serial I/O units are not supported. Contact Opto 22 Product Support for current information on serial I/O support in ioControl and with SNAP PAC controllers.

Additional OptoControl and ioControl Differences

Note the following important differences between OptoControl and ioControl:

- **PIDs on SNAP Ethernet brains have more features and options than *mistic* PID loops.** PIDs have been divided into their own command groups in ioControl: PID-Ethernet commands and PID-Mistic commands. See Chapter 10 of the *ioControl User's Guide* for more information.

- **ioControl uses a non-zero value to represent true and 0 for false.** This is different from OptoControl, which used -1 for true and 0 for false. This also affects true/false returns on commands such as on/off pulse and latches.
- **Error codes, both returned and queued, are different in OptoControl and ioControl.** See note 4 in [“Notes for OptoControl and ioControl Commands” on page 22](#) for more information on differences in return values.
- **The sign used on quadrature counting is different in *mistic* and SNAP Ethernet brains.** See “Digital Point Commands” in Chapter 10 of the *ioControl User’s Guide* for more information.

2—Preparing the Strategy

Before converting your OptoControl strategy to ioControl, do the following:

- **Back up the current, working OptoControl strategy and subroutines to one or more secure locations.**
- **Check for I/O units configured on more than one serial port.** In the strategy tree, under I/O Units, double-click the I/O Units folder to see a list of all I/O units for the strategy, including Port and Address field. Because the SNAP-PAC-S1 has only one serial I/O port, any duplicate addresses or unique ports must be changed before opening the OptoControl strategy in PAC Control. All serial I/O units must be configured for the same serial port. If these changes are not made, the conversion will stop.
- Once duplicate addresses have been changed (I/O units re-jumpered), before converting the strategy to ioControl, we recommend that you re-jumper and rewire your existing controller to use just one port to verify this configuration works in OptoControl.

NOTE: When additional I/O units are added to a serial port, OptoControl will automatically assign new addresses. For example, if Port 0 has 25 devices on it with the addresses 0 to 24, the next device (the 26th) added to that port will be assigned address 25.

- **Check for any ARCNET I/O units.** ARCNET I/O units, such as the B3000-HA and the SNAP-BRS-HA, are not supported in PAC Project. You’ll need to replace these brains with Ethernet-based or serial brains. In the OptoControl strategy, create new replacement I/O units. Then move I/O points from the old ARCNET units to the new units.
- **Check for external instructions.** In the strategy tree there’s a folder called “Ext. Instruction Files Included.” If this folder is not empty, you’ll need to figure out what the function of those commands is and then implement it using OptoControl commands.

- **Check subroutines.** OptoControl and ioControl subroutines support different parameters. Any unsupported parameters passed to a subroutine will need to be handled differently.

OptoControl Subroutine Data Types Supported in ioControl

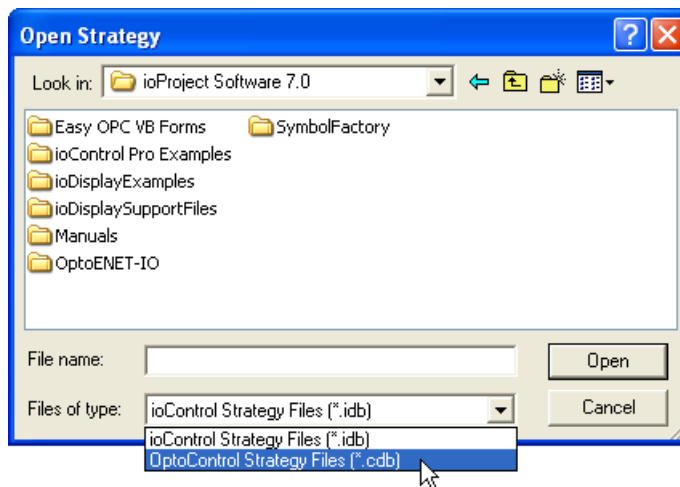
- Numeric variables, literals, and tables
- String variables, literals, and tables
- Communication handles
- I/O points, I/O units, and pointer tables
- Pointer variables as local parameters

OptoControl Subroutine Data Types Not Supported in ioControl

- Unsupported I/O units and I/O points on them, such as G4 or HRD I/O units
- Event/reactions
- Charts
- PID loops

3—Converting the Strategy

1. To open ioControl, from the Windows Start menu select All Programs→Opto 22→ioProject Software→ioControl.
2. In ioControl, select File→Open Strategy.
3. In the Open Strategy dialog box that appears, select “OptoControl Strategy Files (*.cdb)” from the “Files of type” drop-down menu.

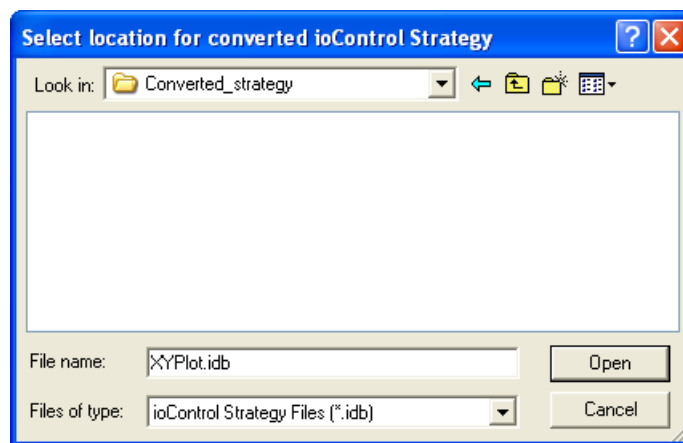


4. Navigate to the OptoControl strategy file you want to convert, select it, and click Open.
 - If there are no problems converting the strategy, a message box will prompt you to choose a location where the converted strategy will be saved. Continue to [step 5](#) on [page 10](#).
 - If there is a problem that prevents the OptoControl strategy from being converted, the conversion process stops. Read the error message describing the problem, modify the strategy as needed, and repeat [step 4](#).

If the conversion process stops while converting the OptoControl strategy, one of the following error messages will appear:

Error message	Description
Cannot open from OptoControl version <versionnumber>.	ioControl can only open OptoControl files saved in OptoControl R4.1. Contact Opto 22 Product Support if you do not have this version of the software.
I/O unit <unitname> is using ARCNET or a local port. Strategy cannot be converted.	Review the information in “1—Assessing Major Differences in Features” on page 6 and “2—Preparing the Strategy” on page 8 for workarounds or other approaches.
Cannot convert from a strategy that uses more than one serial port.	Review the information in “2—Preparing the Strategy” on page 8 for workarounds or other approaches.

5. Click OK to close the message box, and in the dialog box that opens, select a location to save the converted strategy and then click Open.



The OptoControl strategy is now converted to ioControl.

4—Saving Strategies to Flash

The converted OptoControl strategy will open in ioControl

If you are currently using M4-series controllers and are now using a SNAP-PAC-S1 for your strategy, be sure that you save your strategy to the SNAP PAC's flash memory after downloading.

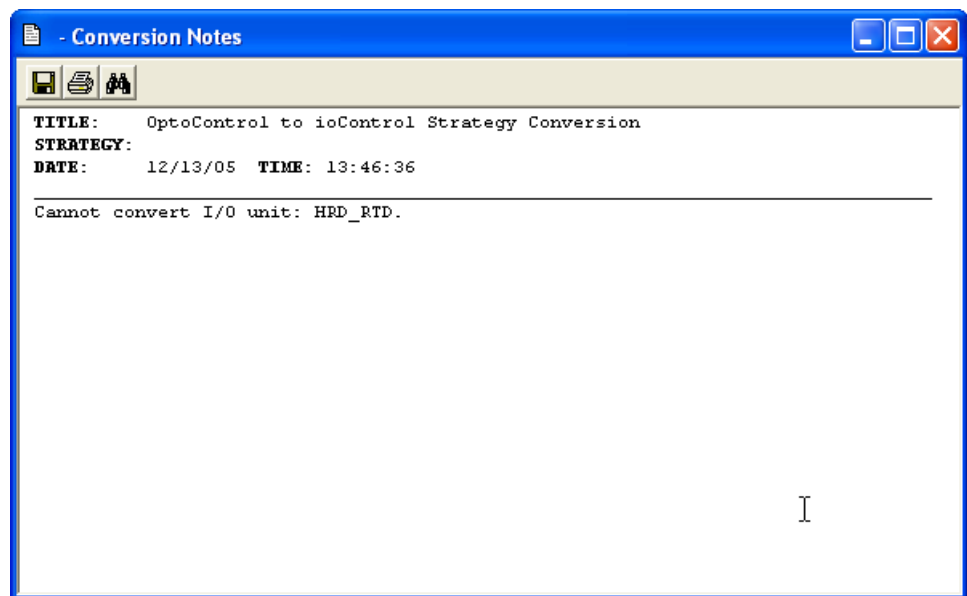
On the SNAP PAC controller, persistent variables and variables initialized on download are automatically saved to battery-backed RAM, but strategies are not. The SNAP PAC controller has 7.5 MB available in flash memory for strategy storage. See the *ioControl User's Guide* (form 1300) for instructions.

4—Followup

The converted OptoControl strategy will open in ioControl. Before using the strategy, you must review and correct conversion errors, test-compile any OptoScript blocks, and finally save the strategy to the controller's flash memory.

Reviewing Conversion Errors

If errors occur that do not stop the conversion process, they are listed in the error log file "Conversion_Notes.rtf". The log file is saved in the same directory as the converted strategy, and opens in its own ioControl window.



Check each message listed in the error log against the list in "[Conversion Error Log Messages](#)" on page 13 and correct the converted strategy as needed.

Test Compiling OptoScript Blocks

Because OptoScript blocks in the OptoControl strategy remain unchanged through the conversion process, you'll need to check them for necessary command and error handling

changes. If OptoScript commands are different in OptoControl and ioControl, the OptoScript block will not compile.

To check OptoScript block for any needed changes, click the Test Compile button in each OptoScript block; if it does not compile, the exact lines of code that need to be updated will be listed. Click the error description to jump directly to the line of code that needs modification. See [page 16](#) for a list of OptoControl commands and ioControl equivalents. Also remember that any PID command used in an OptoControl strategy now has "mistic" in its command name in ioControl to distinguish it from Ethernet PID commands.

Saving Strategies to Flash

IMPORTANT: *If you are currently using M4-series controllers and are now using a SNAP-PAC-S1 for your strategy, be sure that you save your strategy to the SNAP PAC's flash memory after downloading.*

On the SNAP PAC controller, persistent variables and variables initialized on download are automatically saved to battery-backed RAM, but strategies are not. The SNAP PAC controller has 7.5 MB available in flash memory for strategy storage. See the *ioControl User's Guide* (form 1300) for instructions.

Conversion Error Log Messages

Error Message	Description
<p>Cannot convert instruction <instructionname> in chart <chartname>, block <blocknumber>.</p>	<p>An instruction used in a chart in the OptoControl strategy isn't available in ioControl. An invalid instruction is turned into a "Comment (OptoControl Conversion Issue)" instruction. This instruction's description string contains the name of the old instruction and the parameters sent.</p>
<p>Cannot convert subroutine <subname>, it contains pointer variables.</p>	<p>A subroutine uses pointers as internal variables, which ioControl doesn't support. All other subroutine variable types used in OptoControl are allowed in ioControl.</p>
<p>Cannot convert subroutine <subname>, it uses invalid parameter types.</p>	<p>OptoControl allows many subroutine parameter types to be used that ioControl does not support. These subroutine parameter types will not convert to ioControl:</p> <ul style="list-style-type: none"> • Analog Event Reaction • B100 Digital Multifunction I/O Unit • B200 Analog Multifunction I/O Unit • B3000 SNAP Analog • B3000 SNAP Digital • B3000 SNAP Mixed I/O • Chart • Digital Event/Reaction • Event/Reaction Group • G4 Analog Multifunction I/O Unit • G4 Digital Local Simple I/O Unit • G4 Digital Multifunction I/O Unit • G4 Digital Remote Simple I/O Unit • HRD Analog Current Output I/O Unit • HRD Analog RTD Input I/O Unit • HRD Analog Thermocouple/mV Input I/O Unit • HRD Analog Voltage Output I/O Unit • HRD Analog Voltage/Current Input I/O Unit • Local Simple Digital Input • Local Simple Digital Output • PID Loop • Pointer Table • SNAP Digital 64 • SNAP Remote Simple Digital • SNAP Simple 64 • TPO
<p>Cannot convert instruction <instructionname> in subroutine <subname>, block <blocknumber>.</p>	<p>An instruction used in a subroutine in the OptoControl strategy isn't available in ioControl. An invalid instruction is turned into a "Comment (OptoControl Conversion Issue)" instruction. This instruction's description string contains the name of the old instruction and the parameters sent.</p>

Error Message	Description
<p>Cannot convert I/O unit: <unitname>.</p>	<p>An I/O unit definition used in the OptoControl strategy isn't available in ioControl. The following I/O units are not supported in ioControl:</p> <ul style="list-style-type: none"> • G4 Digital Local Simple (G4D16LS) • HRD Analog RTD Input (G4AIRTD) • HRD Analog Thermocouple/mV Input (G4AITM) • HRD Analog Voltage/Current Input (G4AIVA) • HRD Analog Current Output (G4AOA) • HRD Analog Voltage Output (G4AOV) • Motion Servo and Stepper boards
<p>Cannot convert analog point: <pointname>.</p>	<p>An analog I/O point on an HRD (high density) I/O unit will not be available because this I/O unit cannot be converted into ioControl. The following analog inputs and outputs on an HRD I/O unit cannot be converted:</p> <p>Inputs</p> <ul style="list-style-type: none"> • Type B Thermocouple (Note that SNAP-AITM2: Type B Thermocouple will convert.) • Type C Thermocouple • Type D Thermocouple • Type E Thermocouple • Type G Thermocouple • Type J Thermocouple • Type K Thermocouple • Type N Thermocouple • Type R Thermocouple • Type S Thermocouple • Type T Thermocouple • 0 - 100 mV (Scalable) • -50 - +50 mV • -150 - +150 mV • 0 - 20 mA • 4 - 20 mA (Scalable) • 4 - 20 mA • 0 - 1.25 VDC • 0 - 5 VDC • 0 - 10 VDC (Scalable) • 0 - 10 VDC • -5 - +5 VDC • -10 - +10 VDC

Error Message	Description
	<ul style="list-style-type: none"> • 0 - 12500 Ohms (Scalable) • 100 Ohm Pt: Alpha=385 3-wire • 100 Ohm Pt: Alpha=385 2/4-wire • 120 Ohm Ni: Alpha=672 3-wire • 120 Ohm Ni: Alpha=672 2/4-wire • 604 Ohm NiFe: Alpha=518 3-wire • 604 Ohm NiFe: Alpha=518 2/4-wire • 100 Ohm Pt: Alpha=3923 3-wire SAMA • 100 Ohm Pt: Alpha=3923 2/4-wire SAMA • 100 Ohm Pt: Alpha=3928 3-wire ITS90 • 100 Ohm Pt: Alpha=3928 2/4-wire ITS90 • 100 Ohm Pt: Alpha=391 3-wire OIML • 100 Ohm Pt: Alpha=391 2/4-wire OIML • 100 Ohm Pt: Alpha=3916 3-wire JIS • 100 Ohm Pt: Alpha=3916 2/4-wire JIS • 100 Ohm Pt: Alpha=3916 3-wire JEMIMA • 100 Ohm Pt: Alpha=3916 2/4-wire JEMIMA • 100 Ohm Pt: Alpha=391 3-wire GOST • 100 Ohm Pt: Alpha=391 2/4-wire GOST <p>Outputs</p> <ul style="list-style-type: none"> • 0 - 5 VDC • 0 - 10 VDC (Scalable) • 0 - 10 VDC • -5 - +5 VDC • -10 - +10 VDC • 0 - 20 mA • 4 - 20 mA (Scalable) • 4 - 20 mA
Cannot convert digital point: <pointname>.	A digital I/O point will not be available if the I/O unit for that point was not converted into ioControl.
Cannot convert PID Loop: <pidname>. I/O Unit was not converted.	A PID loop will not be available if the I/O unit for that PID loop was not converted into ioControl.
Cannot convert Event/Reaction: <ername>. I/O Unit was not converted.	An event/reaction (E/R) will not be available if the I/O unit for that E/R was not converted into ioControl.
Strategy has external action definitions which won't be converted.	External actions are not supported in ioControl.
Strategy has external condition definitions which won't be converted.	External conditions are not supported in ioControl.
Added new numeric variable: <variablename> for returning subroutine status.	This message will appear for any OptoControl strategy that includes subroutines. In ioControl, all subroutines return a status parameter. Because this parameter does not exist in OptoControl, a new numeric variable "SubroutineReturnStatus" is created for the subroutine status parameter.

Error Message	Description
A subroutine and chart are both named <suband-chartname>. This will cause problems when you run the strategy.	Subroutines and charts in an OptoControl strategy can share the same name (for example, a chart and a subroutine can both be named Valve_Control). In ioControl, subroutines and charts cannot share the same name. You must change the name of either the chart or the subroutine before running the converted strategy.

Test Compiling OptoScript Blocks

Because OptoScript blocks in the OptoControl strategy remain unchanged through the conversion process, you'll need to check them for necessary command and error handling changes. If OptoScript commands are different in OptoControl and ioControl, the OptoScript block will not compile.

To check OptoScript block for any needed changes, click the Test Compile button in each OptoScript block; if it does not compile, the exact lines of code that need to be updated will be listed. Click the error description to jump directly to the line of code that needs modification. See [page 16](#) for a list of OptoControl commands and ioControl equivalents. Also remember that any PID command used in an OptoControl strategy now has "mistic" in its command name in ioControl to distinguish it from Ethernet PID commands.

OptoControl and ioControl Commands

Most commands convert automatically. However, some OptoControl commands are different or unavailable in ioControl, and you'll need to change your strategy as noted in the following table. For each OptoControl command, the comparable ioControl command and any applicable notes are listed. Table footnotes follow on [page 22](#).

OptoControl Command	Comparable ioControl Command(s)	Notes
Accept Session On TCP Port	Accept Incoming Communication	11
ARCNET Connected?	n/a	5
ARCNET Message Address Equal to?	n/a	5
ARCNET Node Present?	n/a	5
Calculate & Store Strategy CRC	n/a	This command is not needed in ioControl. ioControl includes options for backing up strategies and saving them on the PC or controller. In ioControl, see Strategy Options under the File menu.
Characters Waiting at Serial Port?	Get Number of Characters Waiting	11
Clear I/O Unit Interrupt	n/a	2
Clear PC Byte Swap Mode (ISA Only)	n/a	3

OptoControl Command	Comparable ioControl Command(s)	Notes
Clear Quadrature Counter	Clear Counter	1—The ioControl command can access quadrature points.
Clear Receive Buffer	Use "Get Number of Characters Waiting" and "Receive N Characters" to achieve the same result. For TCP communication handles, use "Clear Communication Receive Buffer."	11
Close Ethernet Session	Close Communication	11
Configure I/O Unit	n/a	In ioControl, I/O units are automatically configured when an I/O unit becomes enabled.
Configure Port	Set Communication Handle Value	11
Configure Port Timeout Delay	Set Communication Handle Value	11—For individual serial I/O units, timeout values can be changed from the 1 sec. default. Timeout values can also be changed for FTP communication handles. TCP timeouts have the following default values: <ul style="list-style-type: none"> • Open: 4 sec. • Selects: 1 sec. • (all other timeouts): 10 sec.
CTS Off?	n/a	10
CTS On?	n/a	10
Disable Communication To Analog Point	Disable Communication to Point	1
Disable Communication To Digital Point	Disable Communication to Point	1—Does not convert if parameter is a pointer or a simple digital point.
Disable Interrupt On Event	n/a	2
Enable Communication To Analog Point	Enable Communication to Point	1
Enable Communication To Digital Point	Enable Communication to Point	1—Does not convert if parameter is a pointer or a simple digital point.
Enable Interrupt On Event	n/a	2
Ethernet Session Open?	Communication Open?	11
Generating Interrupt?	n/a	2
Get & Clear Digital I/O Unit Latches	n/a	8
Get & Clear Digital-64 I/O Unit Latches	n/a	8
Get & Clear Quadrature Counter	Get & Clear Counter	1
Get & Clear Simple-64 I/O Unit Latches	n/a	8
Get Active Interrupt Mask	n/a	2
Get Address Of I/O Unit Causing Current Error	Get Name of I/O Unit Causing Error	4

OptoControl Command	Comparable ioControl Command(s)	Notes
Get Analog Lower Clamp	n/a	1
Get Analog Upper Clamp	n/a	1
Get ARCNET Destination Address On Port	n/a	5
Get ARCNET Host Destination Address	n/a	5
Get ARCNET Peer Destination Address	n/a	5
Get Controller Address	"Get Control Engine Address" and "Convert IP Address String to Integer 32" (if needed)	1
Get Default Host Port	n/a	ioControl controllers do not have jumpers.
Get Digital I/O Unit As Binary Value	Get I/O Unit As Binary Value	1—Does not convert if original "Put In" parameter was a float variable.
Get Digital I/O Unit Latches	n/a	8
Get Digital-64 I/O Unit As Binary Value	Get I/O Unit As Binary Value	1
Get Mixed I/O Unit As Binary Value	Get I/O Unit As Binary Value	1
Get Simple-64 I/O Unit As Binary Value	Get I/O Unit As Binary Value	1
Get Digital-64 I/O Unit Latches	n/a	8
Get Ethernet Session Name	Get Communication Handle Value	11
Get Number Of Characters Waiting On Ethernet Session	Get Number of Characters Waiting	11
Get Number Of Characters Waiting On Serial Or ARCNET Port	Get Number of Characters Waiting	5, 11
Get Port Of I/O Unit Causing Current Error	Get Name of I/O Unit Causing Error	4
Get Priority	n/a	6
Get Priority Of Host Task	n/a	6
Get Quadrature Counter	Get Counter	1
Get RTU/M4IO Temperature	n/a	7
Get RTU/M4IO Voltage	n/a	7
Get Simple-64 I/O Unit Latches	n/a	8
Host Task Received a Message?	n/a	9
Interrupt Disabled for Event?	n/a	2
Interrupt Enabled for Event?	n/a	2
Interrupt On Port0?	n/a	2
Interrupt On Port1?	n/a	2
Interrupt On Port2?	n/a	2

OptoControl Command	Comparable ioControl Command(s)	Notes
Interrupt On Port3?	n/a	2
Interrupt On Port6?	n/a	2
IVAL Set Analog From Table	IVAL Move Numeric Table to I/O Unit	8
IVAL Set Quadrature Counter	IVAL Set Counter	1
Low RAM Backup Battery?	n/a	7
Move Analog I/O Unit To Table	Move I/O Unit to Numeric Table	1—Does not convert if parameter is a pointer.
Move Digital I/O Unit To Table	Move I/O Unit to Numeric Table	1
Move Digital I/O Unit To Table Element	Get I/O Unit as Binary Value	1—Use OptoScript to move value(s) to a table element using Table[index] syntax. Example: Int32table[0] = GetIoUnitAsBinaryValue(UIO);
Move Mixed I/O Unit To Table	Move I/O Unit to Numeric Table	1
Move Simple-64 I/O Unit To Table	Move I/O Unit to Numeric Table	1
Move Table Element To Digital I/O Unit	Set Digital I/O Unit From MOMO Masks	1
Move Table To Analog I/O Unit	Move Numeric Table to I/O Unit	1—Does not convert if parameter is a pointer.
Move Table To Digital I/O Unit	Move Numeric Table to I/O Unit	1
Move Table To Simple-64 I/O Unit	Move Numeric Table to I/O Unit	1
Open Ethernet Session	Open Outgoing Communication	11
Read Byte From PC Memory (ISA Only)	n/a	3
Read Byte From PC Port (ISA Only)	n/a	3
Read Word From PC Memory (ISA Only)	n/a	3
Read Word From PC Port (ISA Only)	n/a	3
Receive Character Via Serial Port	"Receive Character" or "Receive N Characters"	11
Receive N Characters Via ARCNET	n/a	5
Receive N Characters Via Ethernet	"Receive Character" or "Receive N Characters"	11
Receive N Characters Via Serial Port	"Receive Character" or "Receive N Characters"	11
Receive String Via ARCNET	n/a	5
Receive String Via Ethernet	"Receive String" or "Receive String Table"	11
Receive String Via Serial Port	"Receive String" or "Receive String Table"	11
Receive Table Via ARCNET	n/a	5
Receive Table Via Ethernet	"Receive Numeric Table"	11

OptoControl Command	Comparable ioControl Command(s)	Notes
Receive Table Via Serial Port	"Receive Numeric Table"	11
Reset Controller	Write Number to I/O Unit Memory Map	This can be done by writing the value 5 to the address 0xF038000 in the controller's memory map. See Opto 22 form 1465 for more information on the memory map.
Set ARCNET Destination Address On Port	n/a	5
Set ARCNET Host Destination Address	n/a	5
Set ARCNET Mode Raw	n/a	5
Set ARCNET Mode Standard	n/a	5
Set ARCNET Peer Destination Address	n/a	5
Set Day Of Week	n/a	The day of the week is automatically calculated based on the date.
Set End-of-message Terminator	Set End-Of-Message Terminator	11
Set I/O Unit Configured Flag	n/a	In ioControl, I/O units are automatically configured when an I/O unit becomes enabled.
Set Mixed I/O Unit From MOMO Masks	Set Mixed I/O Unit from MOMO Masks	1
Set Number Of Retries To All I/O Units	n/a	Retries are set on an I/O unit when the I/O unit is configured.
Set PC Byte Swap Mode (ISA Only)	n/a	3
Set Priority	n/a	6
Set Priority Of Host Task	n/a	6
Set Simple-64 I/O Unit From MOMO Masks	Set Simple-64 I/O Unit From MOMO Masks	1
Start Continuous Square Wave	Start Continuous Square Wave	1 —On Ethernet-based units, new square waves take effect as soon as a Start Continuous Square Wave instruction is sent.
Start Default Host Task	n/a	9
Start Host Task (Ascii)	n/a	9
Start Host Task (Binary)	n/a	9
Start Quadrature Counter	Start Counter	1
Stop Host Task	n/a	9
Stop Quadrature Counter	Stop Counter	1
Suspend Default Host Task	n/a	9
Transmit Character Via Serial Port	Transmit Character	11

OptoControl Command	Comparable ioControl Command(s)	Notes
Transmit Newline Via Serial Port	<ul style="list-style-type: none"> Transmit Character—use twice, once for character 13 (CR), then 10 (LR). Transmit String—use character 13 (CR) as the last character and set the EOM to 10 (LR). 	11
Transmit String Via ARCNET	n/a	5
Transmit String Via Ethernet	Transmit String	11
Transmit String Via Serial Port	Transmit String	11
Transmit Table Via ARCNET	n/a	5
Transmit Table Via Ethernet	Transmit Numeric Table	11
Transmit Table Via Serial Port	Transmit Numeric Table	11
Transmit/receive Mystic I/O Hex String With Checksum	n/a	Checksum is not correctly supported.
Transmit/receive Mystic I/O Hex String With CRC	Transmit/Receive Mystic I/O Hex String	1—See the <i>ioControl Command Reference</i> (form 1301) for additional options for this command.
Transmit/receive Optomux String	Transmit/Receive String	1—The "Transmit/Receive String" command may be used to send a serial Optomux (or other protocol) string using a serial communication handle. B3000 brain users may consider reconfiguring their brain's jumpers for <i>mistic</i> mode. B1 and B2 brain-board users should consider upgrading to an E1 or E2 Ethernet-capable brainboard.
Transmit/receive String Via ARCNET	n/a	5
Transmit/receive String Via Ethernet	Transmit/Receive String	11
Transmit/receive String Via Serial Port	Transmit/Receive String	11
Turn Off RTS	n/a	10
Turn Off RTS After Next Character	n/a	10
Turn On RTS	n/a	10
Write Byte To PC Memory (ISA Only)	n/a	3
Write Byte To PC Port (ISA Only)	n/a	3
Write Word To PC Memory (ISA Only)	n/a	3
Write Word To PC Port (ISA Only)	n/a	3

Notes for OptoControl and ioControl Commands

1. The OptoControl command is directly converted to a comparable ioControl command except as noted.
2. *mistic* interrupts are not supported in ioControl Professional 7.0. This includes the RS-485 link interrupt wires, the interrupts chart, and Event/Reactions related to interrupts. Visit the Opto 22 website (www.opto22.com) or contact Opto 22 Product Support for current information about this feature.
3. Legacy Opto 22 ISA controllers are not supported in ioControl. Consider moving your control logic to SNAP PAC standalone and rack-mounted controllers.
4. Error handling commands are different and more extensive in ioControl.

IMPORTANT: *Keep in mind that the status and error codes that are returned by many ioControl commands and that appear in the error queue often differ from the comparable command in OptoControl. Error handling charts and code may need to be modified accordingly.*

If you are currently checking for errors by error number, Opto 22 recommends that you check instead for 0 (success) versus non-0 (error).

To search for places in your strategy that check for an exact match, try searching for the "Equal?" condition.

- a. In Configure Mode, choose Edit→Find.
The Find dialog box appears.
- b. Under Search Scope, select Global.
- c. Under Search For, select the Instruction and Condition radio button.
- d. Under Instruction, select Equal?

Check each occurrence for any references to specific error codes that may have changed between OptoControl and ioControl.

5. ARCNET is not supported by ioProject software. B3000-HA brains may be replaced by B3000 (serial) brains or one of several SNAP Ethernet brains without having to change the rack, modules, and field wiring.
6. "Priority" commands are not supported in ioControl. You can make ioProject charts more efficient by careful use of subroutines and "Delay" commands. For more information on this, see the *ioControl User's Guide* (form 1300).
7. M4RTU and M4IO controllers had special maintenance commands which do not apply to ioProject controllers.
8. To simultaneously read multiple values from a brain, or other values not directly supported by built-in ioControl commands, you can construct your own low-level protocol commands to send directly to the I/O unit. This method is more difficult and error-prone than using built-in ioControl commands for accessing I/O, but may be useful in certain cases such as speed-sensitive applications.

For example, you can construct a *mistic* command to read multiple latches (command 'S'), or to get analog lower/upper clamp values (command 'r'). *mistic* commands constructed manually within a strategy can be sent using the ioControl command "Transmit/Receive Mistic I/O Hex String". For more information, see the *Mistic Protocol User's Guide* (Opto 22 form 270). For Ethernet-based brains, multiple addresses in the memory map can be read at once using the ioControl commands in the "I/O Unit – Memory Map" group (for example, "Read Numeric Table from I/O Unit Memory Map"). See the *OptoMMP Protocol Guide* (Opto 22 form 1465) for information about the memory map.

You can also read or set multiple values conveniently in ioControl by looping through similarly named and sequential variables. This is done by using pointers and the command "Get Pointer From Name." For example, to use a "for loop," create the point names (for example: "Point0", "Point1", "Point2," etc.) and then use the command "Get Pointer From Name" to update a pointer which you can then use to get/set latches for that point in the loop.

9. ioControl does not support manually manipulating the host task(s). Only Ethernet is used for host communications, supporting up to 256 sessions on the host port for SNAP PAC S-series controllers (the SNAP-LCE controller and SNAP Ultimate brains support 32 sessions). Visit the Opto 22 website (www.opto22.com) or contact Opto 22 Product Support for current information about this feature.
10. RS-232 handshaking using Communication Handles is not supported in ioControl. Visit the Opto 22 website (www.opto22.com) or contact Opto 22 Product Support for current information about this feature.
11. Rather than referencing ports by number, data can now be transmitted and received using variables called Communication Handles. More generic open, close, transmit, receive, and other commands can be used with a variety of Communication Handle types, including TCP, serial, local files, and FTP. For more information, see Chapter 10, "Communication Commands," in the *ioControl User's Guide* (Opto 22 form 1300).

Converting OptoDisplay Projects to ioDisplay

An OptoDisplay project is converted into ioDisplay using a separate conversion utility.

NOTE: Remember that ioDisplay Professional can access a FactoryFloor controller running an OptoControl strategy. Specifically, if a SNAP-LCM4 controller using an M4SENET-100 Ethernet adapter card is running an OptoControl strategy, an ioDisplay project running on a PC can connect to that controller.

1—Preparing the Project

1. **If you have not already done so, back up the current, functioning OptoDisplay project to one or more separate locations.**

2. Confirm that the OptoDisplay project is saved in OptoDisplay R4.1. If you have an older version of OptoDisplay, contact Opto 22 Product Support.
3. Confirm that the FactoryFloor controllers used by the OptoDisplay project are configured on the PC used for converting to ioDisplay.

2—Converting the Project

1. From the Start menu, select Opto 22→ioProject Software→Tools→OptoDisplay Converter.
You can also select File→Convert OptoDisplay Project in ioDisplay Configurator when no project is loaded.
2. From the File menu of the OptoDisplay Converter, select Open.
3. In the dialog box, navigate to the location of the OptoDisplay project to be converted.
4. Select the OptoDisplay project (the .MMI file) and click Open.
The OptoDisplay project is converted. If any errors occur, a warning will be displayed.
5. In the dialog box, navigate to the folder where you want to save the new ioDisplay project and then click OK.

3—Followup

After converting the OptoDisplay project to ioDisplay, open and save the converted file, and then update the list of controllers which the project references.

1. Open ioDisplay Configurator.
IMPORTANT: Do not try to open the converted project in ioDisplay Runtime.
2. Select File→Open. In the Open File dialog box, navigate to the location where the converted file was saved.
3. Select the file and click OK.
4. If the project's windows do not appear in ioDisplay Configurator, go to Windows→Open, select the windows you want to open, and then click OK.
5. Select Configure→Control Engine(s).
6. If the project uses multiple controllers, note the order of the FactoryFloor controllers listed in the Control Engines dialog box.
You will need to replace these controllers with ioProject controllers in the same order.
7. Select each controller in the list and click Delete.
8. Now click Add and select the ioProject controller(s) to be used.

Note that you will need to locate the strategy running on the ioProject controller; this is described in “Chapter 5, Configuring Control Engines and Tags” in the *ioDisplay User’s Guide* (form 1302).

If the project uses multiple controllers, each controller must be added in the original order you noted in [step 6](#).

- 9.** Click OK to close the Control Engines dialog box.
- 10.** Select File→Save to save the converted project.

The converted OptoDisplay project is now ready for use.