

FactoryFloor 4.0 Release Notes

Welcome to FactoryFloor 4.0

The new release of Opto 22 FactoryFloor[®] makes it easier than ever to finish your automation projects faster, cheaper, and better than before. This document describes new features and improvements, and provides notes you'll need to use FactoryFloor 4.0.

See ["What's New"](#) below for descriptions of the improvements and enhancements that have been made to applications in the FactoryFloor software suite. See ["Special Notes"](#) on page 3 for important information on using FactoryFloor applications.

What's New

OptoControl™

OptoScript™

The major new addition to OptoControl is OptoScript, an optional programming language that provides an alternative way to program within OptoControl's flowchart environment. OptoScript is a procedural type of computer language similar to Pascal, C, or BASIC, and it can be used within any OptoControl strategy to replace or supplement standard OptoControl commands. Since functions in OptoScript are provided by commands that are almost identical to standard OptoControl commands, you have the same complete range of functions.

You use OptoScript code in its own hexagonal flowchart block and develop it in an easy-to-use editor. For several common programming tasks OptoScript can reduce the number of intermediate variables and commands needed, simplifying flowcharts and making them easier to read. OptoScript is especially useful for:

- math expressions
- string handling
- complex loops

- case statements
- conditions
- combining math expressions, loops, and conditions.

The *OptoControl User's Guide* includes complete instructions for using OptoScript, as well as comparisons with standard programming languages and equivalent standard OptoControl commands. The *OptoControl Command Reference* and online command help provide OptoScript examples and details for individual commands.

See ["Converting Existing OptoControl Code to OptoScript" on page 4](#) for suggestions to help you convert existing code.

Line-by-Line Debugging

OptoControl now includes line-by-line stepping through your flowchart in the debugger, a feature requested by customers. With the new choice of Minimal Debug or Full Debug, you can step through your strategy block by block, or step inside blocks and debug line by line. For Action or Condition blocks, you step one instruction at a time; for OptoScript blocks, you step through each line of code.

New OptoControl Commands

The following new commands have been added to OptoControl.

- Disable Communication to All I/O Points (Simulation action)
- Enable Communication to All I/O Points (Simulation action)
- Communication to All I/O Points Enabled? (Simulation condition)
- Communication to All I/O Units Enabled? (Simulation condition)
- Move 32 Bits (Logical action)
- Get Default Host Port (Controller action)
- Move Table to Table (Miscellaneous action)

See the *OptoControl Command Reference* or online help for details on each command.

Other Changes

- Support for Microsoft® Windows® 2000 has been added.
- In response to customer requests, the Find dialog box now includes a Block ID column, so you can identify a block by its number as well as its name.
- The look of blocks in a chart being debugged has changed to help differentiate the step highlight of a paused chart from the step highlight of a chart that is running. The step highlight on a block whose code is not being executed is the normal red color. While the code in the block is being executed, however, the highlight pulses green.
- Numeric tables are processed much faster in the debugger.

OptoDisplay™

Operator Logging

With this new security feature in OptoDisplay, you can log key operator actions at runtime. These actions must be associated with on-screen graphics and include changing process values or discrete items, downloading or uploading recipes, launching applications, and detaching or attaching controllers. If the operator is allowed to modify alarm point values, these modifications can also be logged. These operator actions are tracked in a log file.

Scanner Information Files

OptoDisplay 4.0 can generate diagnostic scanner information files for debugging throughput problems with the OptoDisplay scanner. A scanner information file is a text file that contains information about the scan groups, refresh groups, and OptoDisplay tags that are being scanned by controllers. The file is created when OptoDisplay Runtime runs an OptoDisplay project. Used along with a list of dynamic attributes generated in OptoDisplay Configurator, a scanner information file can help you locate bottlenecks and inefficiencies in an OptoDisplay project.

Special Notes

System Requirements

System requirements have changed: Here's what you need to install and run FactoryFloor 4.0:

- At least the minimum processor required for your version of Windows (400 MHz Pentium II or better recommended)
- Microsoft Windows 95, Windows 98, Windows ME, Windows 2000, or Windows NT 4.0 (with latest Microsoft service pack installed)
- RAM as shown in the following table:

	Windows 95/ Windows 98	Windows ME	Windows NT	Windows 2000
Minimum	16 MB	32 MB	32 MB	64 MB
Recommended	64 MB	64 MB	64 MB or more	128 MB

- At least 80 MB of available hard drive space.

Upgraded Controller Firmware for FactoryFloor 4.0

FactoryFloor 4.0 requires the latest Opto 22 controller firmware, which is also version 4.0. The new firmware is included with FactoryFloor and can also be downloaded from the Opto 22 Web site. See Chapter 4, "Working with Controllers," in the *OptoControl User's Guide* for instructions to download new firmware to the controller.

Because the new firmware uses more controller RAM than previous versions, large strategies may not have enough room in controller memory. If you have large strategies to run and your controller memory is upgradable, you may need to increase RAM. If the controller memory is fixed, contact Opto 22 Product Support for suggestions.

IMPORTANT: *Before you open any existing strategy in OptoControl 4.0, make a backup copy of the strategy. Having a backup copy will keep your options open in case there is insufficient memory in your controller.*

FactoryFloor 4.0 Controller Support

Please note the following advisories for FactoryFloor 4.0:

- OptoRuntimePC is not included or supported in FactoryFloor 4.0. If you need OptoRuntimePC, continue using your current version of FactoryFloor.
- FactoryFloor 4.0 is not designed for programming Opto 22 SNAP Ultimate I/O™. Instead, use ioControl™ for flowchart-based control (including OptoScript) and ioDisplay™ for a graphic operator interface. More information on SNAP Ultimate I/O, ioControl, and ioDisplay is available on our Web site, www.opto22.com.

New Program File Location

All FactoryFloor applications are now installed in a separate FactoryFloor 4.0 directory on your computer. To start any of these programs, choose Start→Programs→Opto 22→FactoryFloor 4.0 and then the name of the program.

Converting Existing OptoControl Code to OptoScript

You may want to convert some existing OptoControl code into OptoScript code in order to take advantage of OptoScript's strengths in mathematical expressions, string handling, and conditional looping. In some cases conversion may be fairly simple; in other cases it may be complex. Make sure you read the *OptoControl User's Guide* chapter on using OptoScript before you begin.

Back up your strategy before starting so you can go back to it if conversion becomes too difficult. Do not delete existing code until you are sure the new code compiles and runs properly.

To convert single action blocks, follow these steps:


1. Open the action block and select all the instructions. Right-click the selection and choose Copy from the pop-up menu.
2. Open an OptoScript block. Right-click in the editor and choose Paste from the pop-up menu.
3. Using the *OptoControl Command Reference* or online command help, convert each instruction to equivalent OptoScript code. Some instructions will convert to OptoScript commands, and some will use OptoScript's built-in functions, such as addition. Keep comments.
4. Once all instructions are converted, check to see if they can be combined and shortened.

To convert several successive action blocks to a single OptoScript block, follow the same steps but paste the instructions from all the action blocks into the one OptoScript block.

Combining several action, condition, and continue blocks into one OptoScript block is a more difficult conversion. Review the OptoScript chapter in the *OptoControl User's Guide*, paying special attention to the examples that compare standard commands with OptoScript code and the troubleshooting section. Here are some additional tips:

- Break up the original code into small groups of blocks and convert these small pieces one at a time.
- Look for areas that can be directly converted to basic OptoScript control structures, such as if/then/else statements, for loops, while loops, and repeat loops.
- Be very careful with the program's flow. Identify all entry and exit points to a group of code.
- Watch out for code that jumps into the middle of the section being converted. Flowcharts make writing "spaghetti" code very easy. To detangle code, consider adding flags to control program flow. Sometimes just introducing a simple test (probably with an "if" statement) can help.
- Instead of converting, just rewrite the section in OptoScript. Rewriting requires a very good understanding of the original code, which conversion may not require; but rewriting can result in better, more efficient code.

Avoiding Programming Errors with OptoScript

Traditionally, many potential programming errors in OptoControl have been avoided because OptoControl does not allow you to leave a dialog box if the box contains errors. For example, you cannot continue to add commands in an action block if a variable you have used does not exist; you must add the variable before continuing. With OptoScript, however, you are not forced to have compilable code in an OptoScript block before continuing on. We suggest that you test compile the code (using the  button) before closing the OptoScript editor, and fix any compiler errors promptly to avoid programming errors.

Duplicate Variable Names

OptoControl no longer allows variables of different types to have the same name. If you are converting a pre-4.0 strategy that contains variables with duplicate names, OptoControl will check for and rename duplicates.

Duplicates are renamed by adding a suffix to the name. The suffix consists of a type-specific tag and variable's internal ID. Possible type suffixes are listed in the table at right.

For example, suppose a string, a float, and an analog input were all named TEMPERATURE. When converted to 4.0, they would all be renamed to names like these:

TEMPERATURE_str_5
 TEMPERATURE_num_12
 TEMPERATURE_ana_78

If the name with the suffixes is too long (names are limited to 50 characters), the name is truncated to allow room for the suffixes. A log of name changes is created so you can see the changes.

Type	Suffix
numeric variable	_num_
string variable	_str_
pointer variable	_ptr_
numeric table	_ntb_
string table	_stb_
pointer table	_ptb_
I/O unit	_iou_
analog point	_ana_
digital point	_dig_
PID	_pid_
event/reaction	_evr_