

Run and Test Projects

This Chapter discusses procedures for building your applications and running them, either for testing or for production.

Factory Studio has the capability to run multiple projects at the same time, simply by changing the port. Users can run multiple instances of the same project using the Runtime and the Test mode. This can enable development and testing without requiring production to stop.

On this page:

- [Running the Application](#)
- [Using the Diagnostic Tools](#)
- [Module Information](#)
- [Build and Pack Projects](#)
- [Project Version Control](#)
- [Info, Server and Client Namespaces](#)

Running the Application

When your application is ready to try out, you can test it in one of the following ways:

- From **Run > Test**—Lets you run the project in "safe mode," which runs the application with some features protected. Historical, Alarm and Event data are logged to temporary files so they will not affect the production logs, and drivers cannot write any data to the devices.
- From **Run > Startup**—Lets you run the fully operational project, without building the project.

You can run the project using both test and startup at the same time.

For information about using the diagnostic tools available, see "[Using the Diagnostic Tools](#)" below.

To test the application:

Do one of the following:

- Go to **Run > Test**.
- Go to **Run > Startup**.
- Enter or select information, as needed.
- Click either **Run Test** or **Run Startup**.

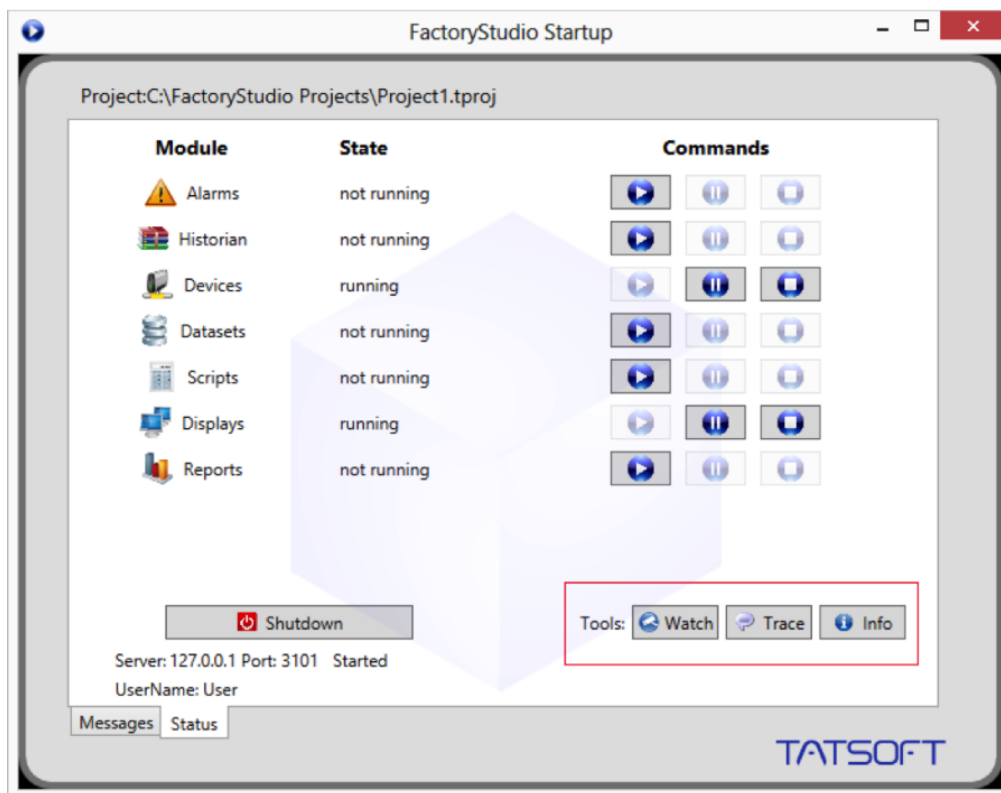
Field	Description
UserName	Enter a valid username to access the application.
Password	Enter the password that corresponds to the username.
Project Server	Read-only. Displays the IP address or name of the computer where the project is based on the configuration on the Server tab.
Port	Displays the port FactoryStudio uses for access. For test, it uses 3201. For startup, it uses 3101. These ports must be open on the server.
PortWA	Displays the port FactoryStudio uses for access with Windows Authentication. For test, it uses 3202. For startup, it uses 3102. These ports must be open on the server.
Use only Windows Authentication	Check if the system can only accept Windows Authentication.
Startup Computer	Read-only. Displays whether the configured server is the local computer or a remote server.
Execution Path	Overrides the default execution path, which is the project file location.
Module Information	Run the Module Information tool.
Property Watch	Run the Property Watch tool.
Trace Window	Run the TraceWindow tool.
Run Modules	Select which Modules execute when running the project.

Status	Field shows current running and connected or disconnected status. - Click Try to Connect to connect to the running application. Required to let changes you make to the project apply immediately to the test or startup runtime. - Click Disconnect if you do not want changes you make to the project to apply immediately to the test or startup runtime. - Click Stop to stop the execution of the application.
Enable Online Configuration	Select to let changes you make to the project apply immediately to the test runtime. You must also be connected to the running project (see Status setting above).

 You can use **Enable Online Configuration** to apply changes on real-time. All changes must be saved before they can appear on screen.

Using the Diagnostic Tools

After starting the project, the Startup window displays the status (running or not running) of each module in your application. In addition, it lets you enable any or all of the following diagnostic tools: PropertyWatch (Watch), TraceWindow (Trace), and ModuleInformation (Info).



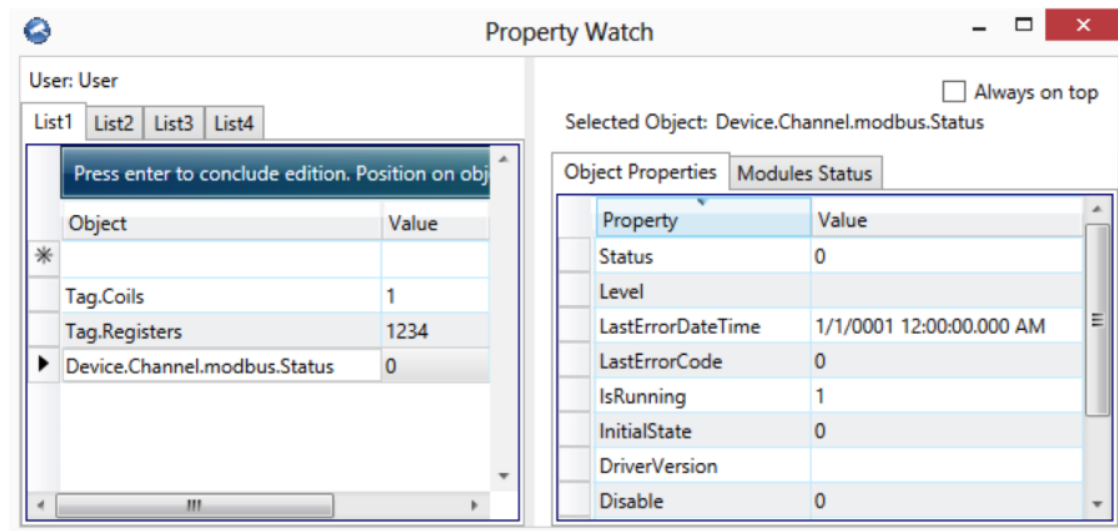
The tools you select will begin running in separate windows as long as your application is running. If you do not see these windows you may find them minimized at the bottom of your screen.

Property Watch

Property Watch is a diagnostic tool used to access tags and internal properties of the system for reading or writing. Type the name of the property in the Object column, and its value, if any, will appear in the Value column. When you type the name of an object type (e.g. "tag") followed by a dot, Intellisense will provide a list of all the available objects. You may select any object from this list, (e.g. "tag.index"), or you may select the type itself (e.g. "tag"). In each case the value, if any, will be given, and all applicable properties with their values will appear in the right hand panel on the "Object Properties" tab.

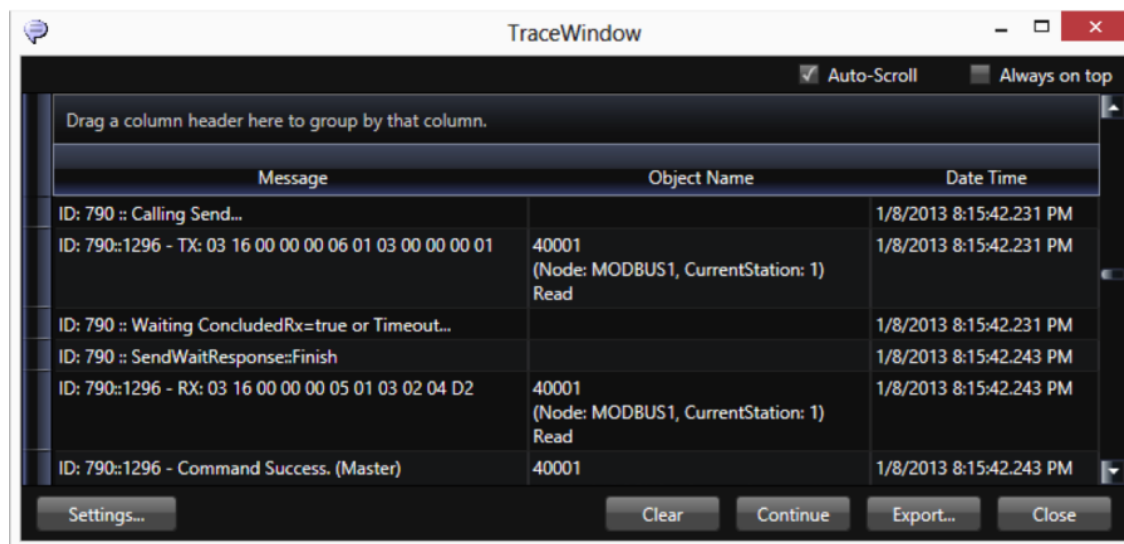
If you select the "Modules Status" tab, you will see the original status panel. While the project is running, you can start or stop individual modules at any time.

You can also change object and property values from the Property Watch screen as long as the changes are permitted.



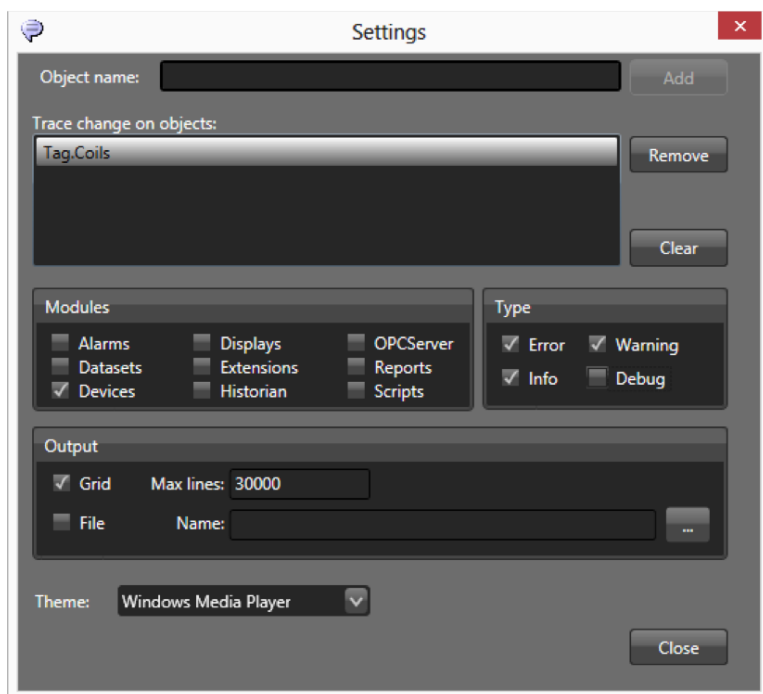
Trace Window

The Trace Window is a tool that displays system messages in a data grid interface. The messages contain status information (based on **Edit > Channels > Settings** column) about reads, writes, unsolicited, TX frames (sent), and RX (received).



i When checking the Devices check box in the Settings, we suggest you enable only the error, info and warning information, not the debug information, otherwise you will create too much data. For ControlLogix devices, it is very important to use this tool, as the system will display the invalid addresses in the configuration.

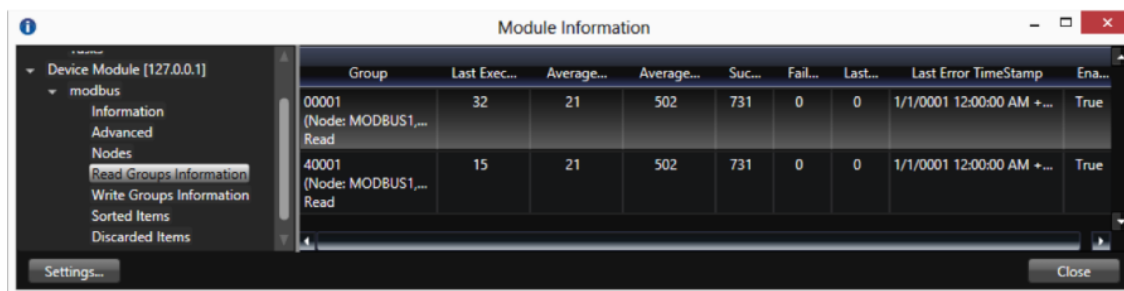
Click **Settings** in the Trace Window to select options for message types and modules, data formats, or to save to file. You can also configure a tag in the Object Name field and click **Add** to select that object to include on the monitoring.



Module Information

Module Information contains information about the operation of the modules. Select the Devices Module and a specific channel to get information about the functioning of the communication channel.

The Read Groups Information provides information about the virtual reading groups, runtime of each item, quantities of readings and readings that have failed, and also reports on the code and date/time of the last error.



To use the Module Information:

- Go to Read Groups Information to look at the number of Successful and Failed communications, to quickly identify the communication blocks.
- If you have systematic errors in all blocks or status codes with negative values, typically it means you cannot access the remote device. Verify if the node address is correct.
- If you have a block with systematic errors, verify the tags and addresses connected with that block. Use the Trace Window with Device information to collect information about those communication errors.
- For some protocols, such as OPC, the Discarded items will show the wrong addresses in the configuration.

When running the Enterprise version in test mode, the system only reads from the field devices, even if you have a configuration to write to field.

You can run the application with online configuration enabled, so you don't need to start and stop the application when modifying the configuration. You can modify PLC addresses, access types, and most of the application and see the real-time the results on your running application. You can use the Startup window or the PropertyWatch "Module Status" tab to start and stop only one module, like Devices, instead of restarting the whole runtime system.

Build and Pack Projects

FactoryStudio is constantly compiling the module you are editing in the background and validating all scripts and displays. If you have not run a full build, the BuildStatus column reflects any warnings or errors found during the background compile process.

Double-click a row with a red X to go to the source of the warning or error. Warnings are informational and do stop the script from running. Errors prevent the specified script from running, but do not affect the whole application. If a script or display has a warning, it will still run.

Periodically, you should run a full build:

- When you have made many changes and you want a full validation and recompile of the whole project.
- When you want to assign a build number to a version.
- When you want to pack the database. When the build executes, the system creates a backup of the current project file. If you want to save the project as it was for this build, rename the backup file.
- When you are getting ready to publish, that is, create the read-only runtime application

To build the application:

- Go to **Run > Build > Messages**.
- If you want to pack the database, select the Pack Database after Build option. Select this option to significantly reduce the size of the project file. The system creates a file with the backup extension, which is the database before the packing. Typically, you want to pack the database every time you run a build.
- If you want to save all displays, select the Save all Displays when Building option. Be sure to use this option if you have made changes to the symbol library. This option applies to all symbol library changes throughout the project.
- Click **Build**.



When checked, a backup of the project connected with that build is automatically created. It is also possible to pack a project without building. To do this just click on the "Pack" button.

Project Version Control

Tracking configuration changes

FactoryStudio provides many ways to help you to keep track of the project configuration changes:

- All configuration tables have the DateCreated and the DateModified information.
- The **Run-Build-History** page shows all Build commands executed in that project. A backup of the project in that state may be available according the user settings.
- The **Info > Track > Tables** page shows all configuration tables, listing the number of rows and if the tables were changed.
- The **Info > Track > Changes** shows all project objects that have been modified. In order to enable or disable that, you must be logged in as Administrator. By default, the object tracking is only enabled after you publish the project, but you can enable it anytime, independently of using or not the publish feature.

Publishing the Project

Publishing the project creates a read-only version of the project.

It is NOT necessary to publish the project to install it for production. In scenarios where you intend to have continuous changes to the project in field, it may be simpler to put the main project file, "tproj", directly on the production computer.

The main benefit of publishing is that the system creates a compact and read-only version of the project file. The file created has the same name of the project file, with the publish version number and the extension "trun". This allows the system to comply with regulated industries.

The typical scenarios that lead to using the publish command are:

- You want to deploy a read-only version of the project, for instance, to be in compliance with certified and regulated environments.
- You want to use the automatic version numbering system. The result of publishing is a ".teng" file that also contains a major (1.0) or minor (0.1) version number as part of the file name. In addition, **Info > Track** helps you manage the files published to the field, including which project build they are.
- You want a smaller footprint and faster loading of the project, for instance on machines, OEM and embedded systems. The "trun" file can be up to 5 to 10 times smaller than the "tproj" file.
- You want to protect the project from modification.



The "trun" file is always read-only, but you may optionally hide the project configuration from the end-user as well. This is an independent option defined in the Security System. If you don't want end-users to see the project configuration, remove the permission of the GUEST user and other users to edit the project before publishing it.

To publish the project:

- Go to **Run > Build > Messages**.
- Select the desired build settings and Click **Build**.
- Go to **Run > Publish**. The Current Project Settings fields show the read-only status of the project.
- Select the Publish Settings you want.
- Click **Publish**.

Info, Server and Client Namespaces

There are a few runtime objects that provide information about running the project.

The **Info** namespace is the main location for runtime status. The main objects on that namespace are:

```
Info.Project: information about the project that is running
Info.License: information on the license
Info.Module: Information and start/stop command to Modules
Info.Trace(): displays a message on the TraceWindow
Info.TestMode: shows if the project is running in Test Mode
```

The **Server** namespace has information about the server computer.

The **Client** namespace has information about each client computer, running a graphical interface.

See "[Namespaces](#)" for the complete programming reference on runtime objects.