Run and Test Projects

This page discusses the procedures for building your applications and running them 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 developing and testing without requiring production to stop.

Running the Application

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

- From **Run > Test**—Run the project in "safe mode," which runs the application while protecting some features. The Historical, Alarm and Event data are logged to temporary files so that they do not affect the production logs. Drivers cannot write any data to the devices.
- From **Run > Startup**—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 available diagnostic tools, 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**.

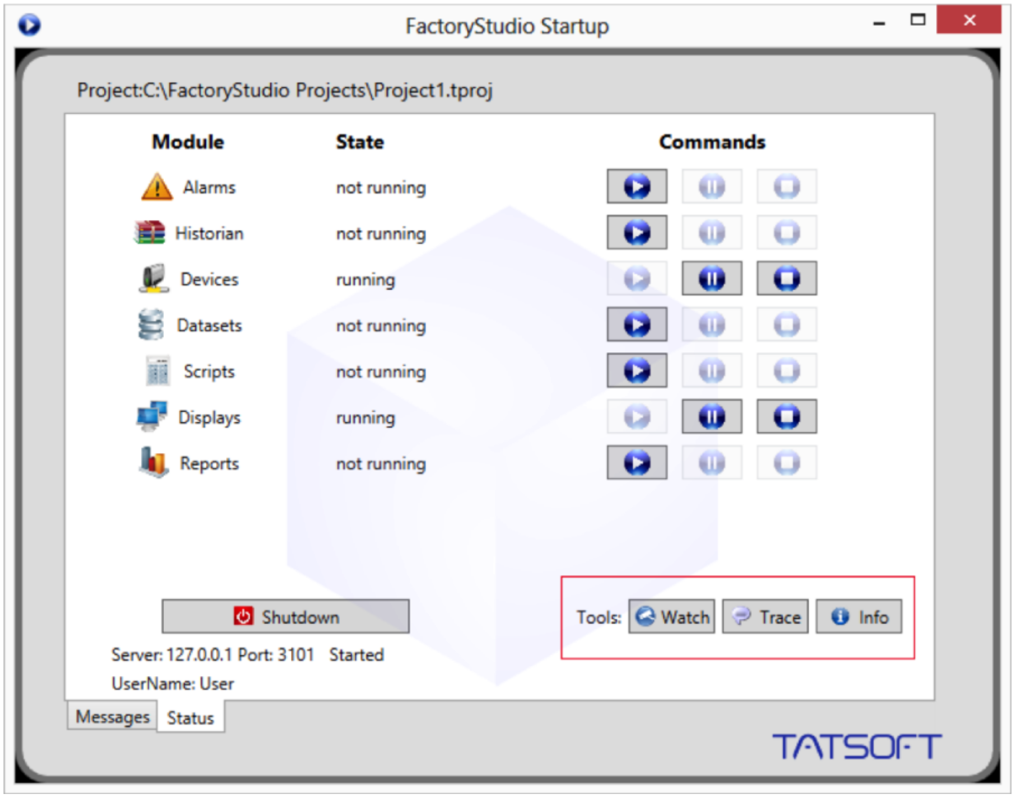
Field	Description
UserName	Enter a valid username to access the application.
Password	Enter the password that corresponds to the username.
Project Server	Read-only. Displays the IP address or the name of the computer where the project is, which is based on the configuration in the Server tab.
Port	Displays the port that 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 that 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 or not 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 are executed when running the project.

Status	Shows the current status of running, connected, or disconnected. • Click **Try to Connect** to connect to the running application. This is required for any of your changes to be immediately applied to the test or the startup runtime. • Click **Disconnect** if you do not want your changes to immediately apply to the test or startup runtime. • Click **Stop** to stop the execution of the application.|
Enable Online Configuration	Select in order for your changes to immediately apply 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 in real-time. All changes must be saved before they can appear on a 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).



Project:C:\FactoryStudio Projects\Project1.tproj

Module	State
Alarms	not running
Historian	not running
Devices	running
Datasets	not running
Scripts	not running
Displays	running
Reports	not running

Server: 127.0.0.1 Port: 3101 Started

UserName: User

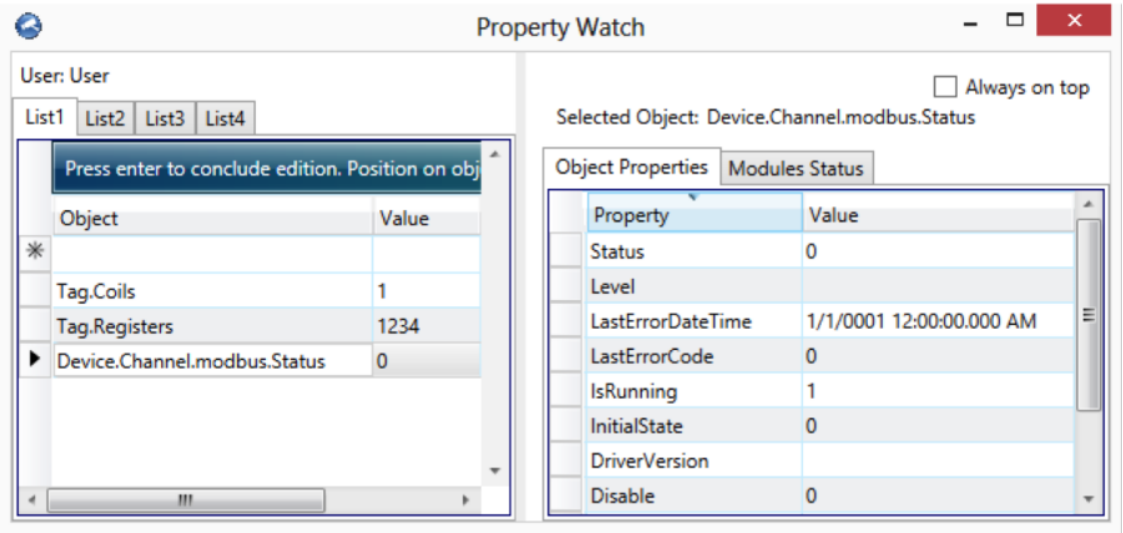
The tools you select will begin running in separate windows as long as your application is running. If you do not see these windows, you may find them minimized at the bottom of your screen.

Property Watch

Property Watch is a diagnostic tool used to access the system's tags and internal properties for reading or writing. Type the name of the property in the Object column, and its value 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 will be provided and all applicable properties and 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 the object and property values from the Property Watch screen as long as the changes are permitted.



User: User

Press enter to conclude edition. Position on obj

Object	Value
Tag.Coils	1
Tag.Registers	1234
Device.Channel.modbus.Status	0

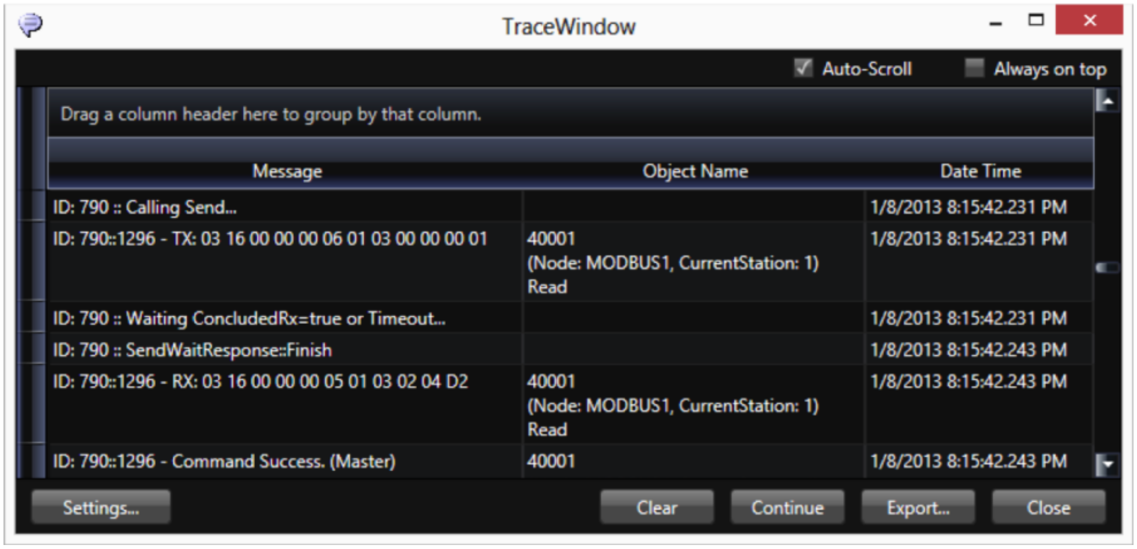
□ Always on top

Selected Object: Device.Channel.modbus.Status

Property	Value
Status	0
Level	
LastErrorDateTime	1/1/0001 12:00:00.000 AM
LastErrorCode	0
IsRunning	1
InitialState	0
DriverVersion	
Disable	0

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).



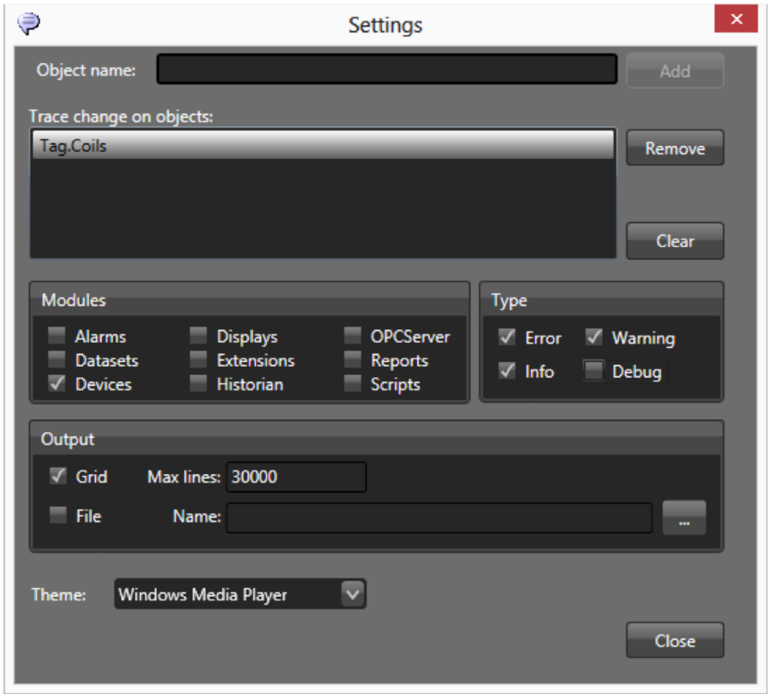
☑ Auto-Scroll □ Always on top

Drag a column header here to group by that column.

Message	Object Name	Date Time
ID: 790 :: Calling Send...		1/8/2013 8:15:42.231 PM
ID: 790::1296 - TX: 03 16 00 00 00 06 01 03 00 00 00 01	40001 (Node: MODBUS1, CurrentStation: 1) Read	1/8/2013 8:15:42.231 PM
ID: 790 :: Waiting ConcludedRx=true or Timeout...		1/8/2013 8:15:42.231 PM
ID: 790 :: SendWaitResponse::Finish		1/8/2013 8:15:42.243 PM
ID: 790::1296 - RX: 03 16 00 00 00 05 01 03 02 04 D2	40001 (Node: MODBUS1, CurrentStation: 1) Read	1/8/2013 8:15:42.243 PM
ID: 790::1296 - Command Success. (Master)	40001	1/8/2013 8:15:42.243 PM

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

In the Trace Window, click the **Settings** to select the options for the message types, 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 in the monitoring.



Object name:

Trace change on objects:

Tag.Coils

Modules

□ Alarms	□ Displays	□ OPCServer
□ Datasets	□ Extensions	□ Reports
☑ Devices	□ Historian	□ Scripts

Type

☑ Error	☑ Warning
☑ Info	□ Debug

Output

☑ Grid Max lines: 30000

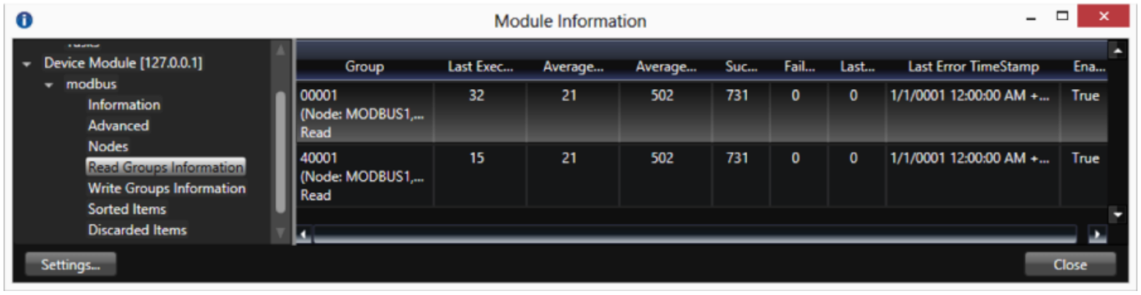
□ File Name:

Theme: Windows Media Player

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, readings that have failed, reports on the code, and date/time of the last error.



Group	Last Exec...	Average...	Average...	Suc...	Fail...	Last...	Last Error TimeStamp	Ena...
00001 (Node: MODBUS1,... Read	32	21	502	731	0	0	1/1/0001 12:00:00 AM +...	True
40001 (Node: MODBUS1,... Read	15	21	502	731	0	0	1/1/0001 12:00:00 AM +...	True

To use the Module Information:

- Go to Read Groups Information to look at the number of Successful and Failed communications and to quickly identify the communication blocks.
- If you have systematic errors in every block or status code that has a negative value, 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 the block. Use the Trace Window with Device information to collect information about the 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 there is a configuration to write to a field.

The online configuration can be enabled when the application is running. This means you do not need to start and stop the application when modifying the configuration. You can modify PLC addresses, access types, and most of the application. You can see the real-time results on your running application. You can use the Startup window or the PropertyWatch "Module Status" tab to start and stop one module, like Devices, instead of restarting the whole runtime system.

Build and Pack Projects

FactoryStudio constantly compiles the module you edit in the background and validates all of the 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.

If a row has a red X, double-click the row to go to the source of the warning or error. Warnings are informational and do not stop the script from running. Errors prevent the specified script from running, but do not affect the whole application. Even 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 to recompile 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 the 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 **"Pack database after build"** to significantly reduce the project file size. The system creates a file with the backup extension, which is the database before the application is packed. You may want to pack the database every time you run a build.
- If you want to save all displays, select **"Verify Symbols and save all Displays"**. 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 that is connected with the build is automatically created. It is also possible to pack a project without building. To do this, click on the "Pack" button.

Project Version Control

Tracking configuration changes

FactoryStudio provides many ways to help you keep track of 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. According to your user settings, you may be able to access a backup version of the project with the settings and values it had when that version was built.
- The **Info > Track > Tables** page shows all of the configuration tables, lists the number of rows, and shows if the tables were changed.
- The **Info > Track > Changes** shows all of the project objects that have been modified. In order to enable or disable this, you must be logged in as an Administrator. By default, object tracking is only enabled after you publish the project, but you can enable it anytime.

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. If the project is expected to have continuous changes when in the field, it is easier 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 real-only version of the project file. The created file has the same name of the project file and it has the publish version number and the extension "trun". This allows the system to comply with regulated industries.

The publish command is usually used in case you want to:

- Deploy a read only version of the project. For instance, to be in compliance with certified and regulated environments.
- 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.
- Have 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.
- Protect the project from modification.

> The "trun" file is always read-only, but you can optionally hide the project configuration from the end-user as well. This is an independent option defined in the Security System. If you do not 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 the runtime status. The main objects on the 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 that runs a graphical interface.

See "Namespaces" for the complete programming reference on runtime objects.